

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Sledování kvality řeči ve VoIP
Speech quality monitoring in VoIP

2011/2012

Bc. Martin Tomeš

Zadání diplomové práce

Student: **Bc. Martin Tomeš**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: **Sledování kvality řeči ve VoIP**
Speech Quality Monitoring in VoIP

Zásady pro vypracování:

1. Metody hodnocení kvality řeči
2. Specifika IP telefonie z pohledu kvality služby
3. Návrh vlastního nástroje pro sledování kvality řeči v IP PBX Asterisk
4. Praktická implementace návrhu
5. Zhodnocení dosažených výsledků

Seznam doporučené odborné literatury:

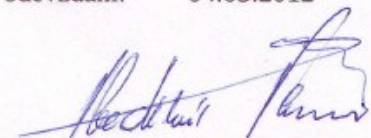
- [1] M. Vozňák, Spojovací systémy. Vydavatel: VŠB-TU Ostrava, 1. vydání, 196 str., 2009, ISBN 978-80-248-1961-7.
- [2] M. Vozňák, Voice over IP. Vydavatel: VŠB-TU Ostrava, 1. vydání, v Ostravě, 176 str., 2008, ISBN 978-80-248-1828-3.
- [3] ITU-T Recommendation P.863, Perceptual objective listening quality assessment, ITU-T Geneva, 2011.
- [4] ITU-T Recommendation G.107, The E-model: a computational model for use in transmission planning. ITU-T Geneva, 2009.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

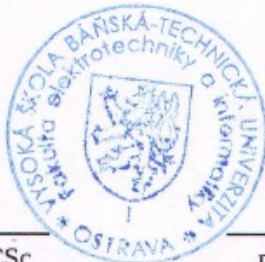
Vedoucí diplomové práce: **doc. Ing. Miroslav Vozňák, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry

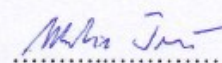


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 4. května. 2012


.....

Poděkování

Rád bych poděkoval panu doc. Ing. Miroslavu Vozňákovi, Ph.D. za veškerou podporu během studia a hlavně odbornou pomoc a konzultace při vytváření této diplomové práce.

Abstrakt

Tato diplomová práce se zabývá návrhem a ověřením funkčnosti systému pro monitorování kvality řeči v IP telefonii. V první části této práce je uveden přehled subjektivních i objektivních metod pro hodnocení kvality řeči včetně podrobnějšího popisu některých důležitějších zástupců. Teoretická část dále obsahuje popis faktorů, které mají největší impakt na kvalitu řeči v IP prostředí.

V praktické části je uveden návrh a zpracování navrženého systému, který staví na metodě E-model, který byl vybrán na základě analýzy v počátečních kapitolách. V další části praktické realizace této diplomové práce je uvedeno srovnávací měření za použití navržené aplikace s měřením metodou PESQ dle doporučení ITU-T P.862, která je považována za etalon v metodách pro měření kvality řeči. Závěrem práce jsou zhodnoceny získané výsledky a navrhnuty možné rozšíření aplikace budoucnu.

Klíčová slova

E-model, PESQ, MOS, kodek, VoIP, Asterisk.

Abstract

This diploma thesis is focused on design and verification of system for voice quality monitoring in IP telephony. First part of this document contains an overview of subjective and objective methods for evaluation of voice quality, including their main delegates and description of their functions. Theoretical part also contains the description of factors, which have direct impact to the voice quality in IP environment.

Practical part of this thesis describes design and functionality of developed system, which is based on objective, non-intrusive method called e-model, which was picked according to analysis from the beginning of this work. Next part of practical realization of this diploma thesis is based on comparative measurement using developed application and measurement obtained using method PESQ corresponding with ITU-T P.862, which is considered to be standard in methods for voice quality evaluation. Evaluation of obtained results and some possible improvements of the system can be found at the end of this work.

Key words

E-model, PESQ, MOS, codec, VoIP, Asterisk.

Obsah

1	Úvod	1
2	Základy IP telefonie	3
2.1	Voice over Internet Protocol	3
2.1.1	RTP protokol	4
2.1.2	RTCP protokol.....	5
2.1.3	RTCP XR.....	6
2.2	SIP protokol.....	8
2.2.1	Žádosti SIP protokolu.....	9
2.2.2	Odpovědi SIP protokolu	10
2.2.3	Struktura SIP hlaviček	10
3	Faktory ovlivňující kvalitu řeči v IP	12
3.1	Ztrátovost paketů	12
3.2	Zpoždění a variabilita zpoždění.....	14
3.3	Echo.....	15
3.4	Výběr kodeku	16
4	Metodiky posuzování kvality řeči	17
4.1	Subjektivní metody hodnocení	18
4.2	Objektivní metody hodnocení	19
4.2.1	Intrusivní testy	20
4.2.2	Neintrusivní testy.....	25
5	Návrh a vývoj aplikace pro měření kvality řeči	29
5.1	Specifikace vyvíjené aplikace	29
5.2	Struktura aplikace.....	30
5.2.1	Web GUI	30
5.2.2	Skripty	31
5.2.3	Quality Monitor	32
5.3	Ovládání aplikace	32
5.3.1	Start/Stop	32

5.3.2	Results	33
5.3.3	Refresh.....	33
5.3.4	Erase	33
5.4	Shrnutí stavu aplikace	34
6	Simulační a srovnávací měření.....	35
6.1	Měření metodou E-model.....	35
6.2	Měření metodou PESQ.....	39
6.3	Prezentace naměřených hodnot	40
7	Zhodnocení výsledků a návrhy rozšíření.....	43
7.1	Vyhodnocení naměřených výsledků.....	43
7.2	Možné rozšíření aplikace.....	44
8	Závěr.....	46
	Použitá literatura	48
	Přílohy	i
	Seznam příloh.....	ii

1 Úvod

Telekomunikační technika a telekomunikace obecně jako obor od svého počátku do dnešní doby prošly značným technologickým vývojem a různými změnami úhlů pohledu. Telekomunikace v dnešní době již nepředstavují pouhé telefonické spojení dvou účastníků, jako tomu bylo na počátku, avšak zahrnují spoustu technologií nových a pokročilejších. Trendem současnosti je tyto technologie vzájemně propojovat a koncovému uživateli nabídnout různá komplexní řešení. Mezi tyto technologie beze sporu patří i přenos hlasu pomocí paketově orientovaných sítí na bázi IP protokolu, jež dostala název Voice over IP - VoIP.

S těmito novými systémy je ovšem také velmi úzce spjata oblast kvality poskytovaných služeb známá jako QoS (Quality of Service). QoS lze souhrnně označit jako soubor různých prostředků ke správě, řízení a monitorování, které mají za úkol přímo stanovit, jak bude uživatel se službou v její konečné fázi spokojen. Jednou z velkých součástí QoS je sledování a posuzování kvality řeči, kterou si lze představit jako jev, kdy na jedné straně máme zdroj signálu (řeči) a na straně druhé přijímač (posluchače) a naším úkolem je určit zřetelnost tohoto reprodukováného signálu a stanovit tak výslednou kvalitu.

VQM (Voice Quality Measurement) neboli měření kvality řeči je disciplína, která nám umožňuje přidání lidské perspektivy z pohledu koncového uživatele oproti jiným, tradičním metodám řízení telekomunikačních sítí. Vhodnost přidání této subjektivní složky je však mnohdy velmi diskutabilní, neboť např. při stanovení kvality jednoho vzorku je možno dosáhnout rozdílných výsledků v závislosti na psychickém rozpoložení, soustředěnosti a znalostech daného posluchače. Dalším podstatným, limitujícím faktorem jsou mnohdy velké náklady na tento způsob měření a jejich zdoluhavost [1], [11].

Moderní telekomunikační technologie (jako např. VoIP) však vyžadují i moderní přístup ke stanovování kvality řeči. Proto vznikly a stále se vyvíjí metody objektivní, které již lidské subjekty přímo nevyžadují a jejich postup je automatizován určitým algoritmem nebo výpočtním modelem.

První část této diplomové práce je zaměřena na základní popis technologie VoIP v současném stavu na bázi SIP protokolu, na kterém je postavena také praktická část této práce. Druhá část se zabývá popisem různých faktorů, které ovlivňují kvalitu hlasu v IP prostředí. Závěr teoretické části je věnován souhrnnému popisu metod stanovení kvality řeči od svého vzniku až po současnost. Praktická část této diplomové práce se zabývá návrhem a realizací jednoduchého monitorovacího nástroje pro sledování kvality řeči a jeho možnosti nasazení na stroji s open source pobočkovou ústřednou Asterisk. Poslední kapitoly jsou věnovány simulačnímu měření a jeho vyhodnocení pro ověření

funkčnosti a správnosti navrhnutého monitorovacího systému. V praktické části je kladen důraz především na ztrátovost procházejících paketů v závislosti na použitém hlasovém kodeku.

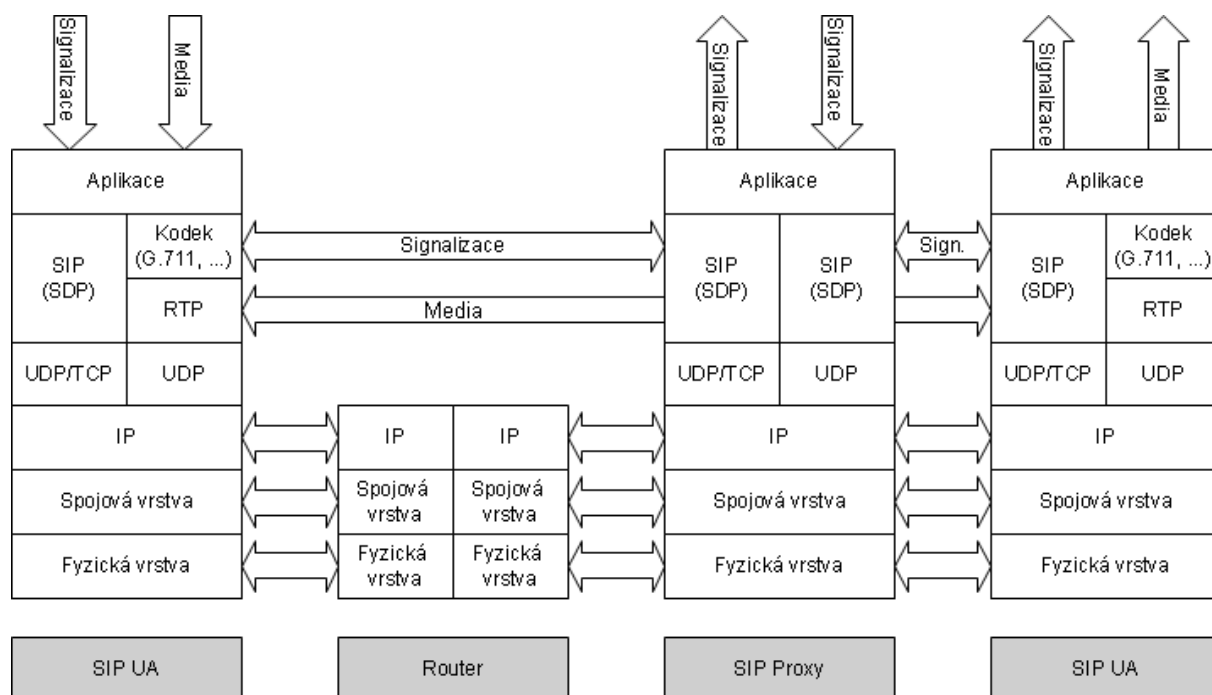
2 Základy IP telefonie

K lepšímu pochopení problematiky stanovení kvality řeči v IP telefonii je dobré znát základy a principy samotné IP telefonie, na kterých je tato technologie postavena. Přenos hlasu přes paketově orientovanou IP síť sebou přináší různé nové aspekty ovlivňující kvalitu řeči oproti "tradičnímu" přístupu k telefonii. Součástí této kapitoly je také základní popis SIP protokolu, který v současné době představuje nejpokročilejší přístup k IP telefonii a také proto, že je využit při praktické části této diplomové práce.

2.1 Voice over Internet Protocol

Označení VoIP v sobě skrývá souhrn protokolů a služeb, které umožňují přenos signalizace a hlasu přes internet nebo paketově přepínanou síť obecně. Pro samotný přenos digitalizovaného hlasu a signalizace rozdělených do jednotlivých paketů se využívá třetí vrstvy referenčního OSI modelu, aby však bylo možné celé spojení sestavit, je potřeba součinnosti protokolů i vrstev vyšších.

Pro přenos hlasu se využívá protokolů RTP (Real-time Transport Protocol) (RFC 1889 – původní specifikace, nyní zastaralá a později nahrazena RFC 3550) a UDP. Pro signalizaci a vyjednání parametrů hovoru slouží v dnešní době především protokoly SIP a SDP. Na obrázku 2.1 je znázorněno zjednodušené rozložení těchto protokolů v RM OSI a vzájemná komunikace na úrovních v přenosovém řetězci. Podrobnější popisy jednotlivých protokolů jsou uvedeny v následujících kapitolách [13].

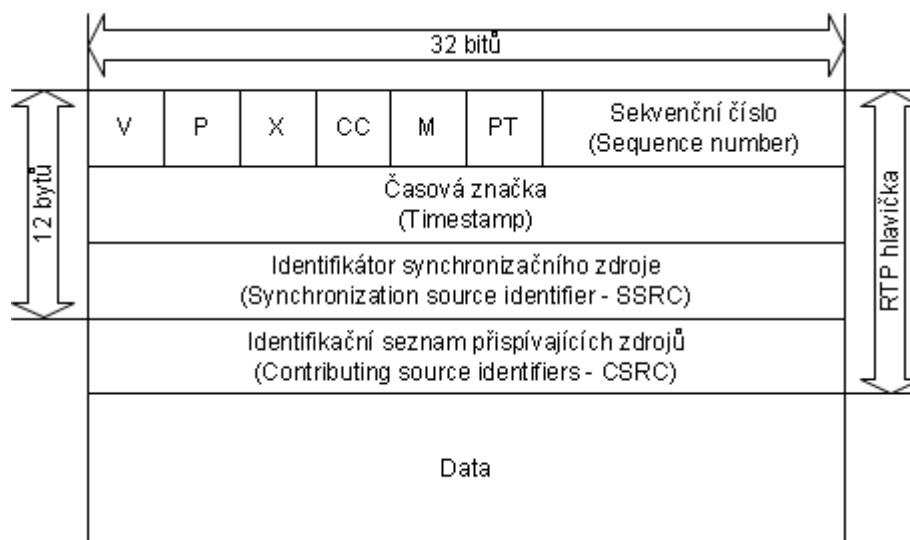


Obr. 2.1 VoIP protokolový stack v RM OSI modelu

2.1.1 RTP protokol

Jak již bylo zmíněno v předchozí kapitole, pro přenos multimediálního obsahu se používá protokol RTP, který využívá transportního protokolu UDP, jehož přenos částečně vylepšuje a přizpůsobuje požadavkům pro přenos hovoru. Z tohoto titulu bývá občas udáván jako další protokol čtvrté, tedy transportní vrstvy. RTP protokol především zajišťuje řazení zaslaných paketů pomocí sekvenčních čísel (Sequence Number) a jejich časové značkování (Timestamp). Další důležitou vlastností, díky které je tento protokol řazen do čtvrté vrstvy OSI modelu, je schopnost multiplexování a demultiplexování digitalizovaného hlasu či videa nadřazeného kodeku. Pomocí protokolu RTP je možné přenášet hovorová data jak mezi dvěma účastníky (unicast), tak i mezi několika účastníky současně (multicast), proto je tento protokol také vhodný pro audio či video konference [2], [19].

Struktura RTP datagramu je znázorněna na obrázku 2.2. Jednotlivé směry RTP relací jsou charakterizovány přiděleným identifikátorem, který se skládá z pole identifikátoru synchronizačního zdroje (SSRC) a dynamicky generovaného UDP portu [19].



Obr. 2.2 Struktura RTP datagramu

Význam jednotlivých polí je v RTP hlavičce následující:

- **V** – určuje verzi RTP protokolu.
- **P** – jedná se o doplnění. Pokud je nastaveno $P = 1$, paket obsahuje na konci jeden či více doplňkových oktetů.
- **X** – jedná se o rozšiřující bit. Pokud je nastaveno $X = 1$, pak je pevné záhlaví následováno právě jedním záhlavím rozšiřujícím s definovanou strukturou.
- **CC** – obsahuje číslo CSRC identifikátoru.

- **M** – je značka, která může mít různé interpretace. Lze ji použít např. jako označení konce rámce v paketovém toku.
- **PT** – znamená payload type a určuje formát užitečného zatížení RTP. Dále stanovuje jeho význam a prioritu pro aplikaci.
- **Sekvenční číslo** – tato hodnota se inkrementuje s každým dalším odeslaným RTP paketem. Příjimač podle tohoto pole určuje případně ztráty a dokáže obnovit souslednost streamu.
- **Časová značka** – vyjadřuje vzorkovací značku prvního oktetu v RTP paketu.
- **SSRC** – náhodně generovaný identifikátor synchronizačního zdroje, který se vytváří vždy jednotlivě pro každou RTP relaci.
- **CSRC** – identifikuje přispívající zdroj z důvodu užitečné informace obsažené v tomto paketu.

2.1.2 RTCP protokol

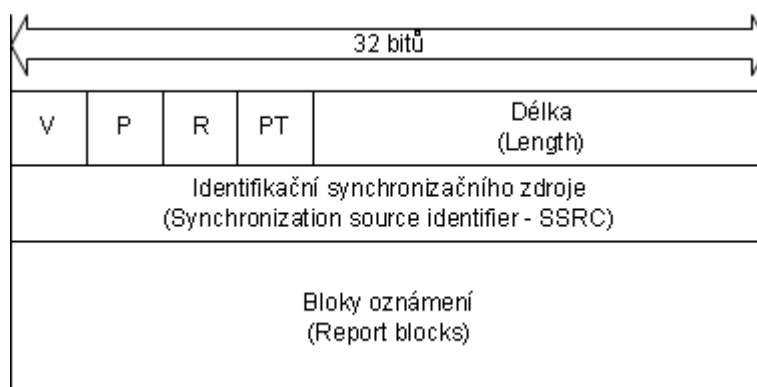
RTP Control Protocol (RFC 3550 [19], RFC 3611 [20]) je protokol doplňující samotný RTP, avšak nenese žádná užitečná multimediální data. Jeho funkcí je především zasílání statistických a přehledových informací o sestaveném spojení a účastnících přenosu. Jak již bylo zmíněno RTCP shromažďuje všechny statistiky přenosu na RTP protokolu, jako jsou např. informace o počtu odeslaných bajtů a paketů, počtu ztracených paketů, rozptylu zpoždění (jitter) a další informace. Data obsažená uvnitř RTCP zpráv mají jistě velký potenciál při nasazení QoS v IP telefonii, avšak opak je pravdou. V současné době většina VoIP ústředěn a ani koncových zařízení s tímto protokolem nedokáže plně a funkčně pracovat, a tak jsou poskytované statistiky častokrát zkreslené a nepravdivé (např. doba zpoždění v řádech několika hodin či dokonce let), či vůbec žádné [2], [19].

V doporučení RFC 3550 [19] je definováno celkem 5 RTCP zpráv s různým obsahem. Jejich struktura je následující:

- **SR (Sender Report)** - soubor všech statistik o přijímaných i vysílaných datech od odesílatelů.
- **RR (Receiver Report)** – soubor všech statistik od příjemců.
- **SDES (Source Description)** – vlastnosti zdrojů RTP komunikace.
- **BYE** – zpráva signalizující ukončení RTCP relace.
- **APP** – zpráva o funkcích specifických jednotlivé aplikaci.

2.1.3 RTCP XR

Rozšíření RTCP zpráv definované v doporučení RFC 3611 [20] z konce roku 2003 je známé jako RTCP XR (Extended Reports). XR pakety jsou složeny z tzv. bloků oznámení (report blocks) a jedním z těchto bloků je i např. VoIP Metrics Report Block, který uceleně informuje o kvalitativních parametrech aktuálního VoIP provozu. Na obrázku 2.3 je uvedena obecná struktura RTCP XR datagramu.



Obr. 2.3 Struktura RTCP XR datagramu

VoIP Metrics blok také poskytuje parametry a data pro monitorování kvality řeči. Aby došlo ke správnému posouzení kvality hlasu je zapotřebí parametrů jako úroveň ztrátovosti paketů, charakter rozložení těchto ztrát (shlukovost) a parametry jiné, na které je posléze aplikován Gilbert-Elliottův model. Tento model v principu pracuje tak, že posuzuje jednotlivé časové úseky a počítá nejen s počtem ztracených paketů, ale také s počtem paketů vyřazených či nezpracovaných důsledkem velkého zpoždění. U těchto paketů je schopen rozlišit jejich charakter rozložení, zda jde o shluk či náhodné rozložení s ojedinělými ztrátami, neboť charakter ztrátovosti hraje podstatnou roli v přesnosti stanovení výsledné kvality. Na obrázku 2.4 je uveden RTCP XR datagram obsahující VoIP Metrics blok [2], [20]. Význam jednotlivých obsažených parametrů je následující:

- **Typ bloku (Block type - BT)** – typ bloku oznámení. Konstanta 7 znamená blok VoIP Metrics.
- **Rezervováno** – 8 bitů rezervováno pro budoucí použití a rozšíření.
- **Délka bloku** – délka bloku je u VoIP Metrics konstantní a má hodnotu 8.
- **SSRC** – pole identifikující zdroj synchronizace.
- **Úroveň ztrát** – hodnota získaná jako počet ztracených paketů od počátku spojení podělená celkovým počtem odeslaných paketů a posléze vynásobena konstantou 256. Udává úroveň ztracených paketů v důsledku vlivů přenosu, nikoliv počet zahozených paketů.

- **Vyřazené pakety** – hodnota získaná jako poměr vyřazených paketů od počátku relace kvůli pozdnímu nebo předčasnému příchodu, podtečení nebo přetečení jitter bufferu u příjemce, k celkovému počtu očekávaných paketů a poté opět vynásobena konstantou 256.
- **Hustota shluku** – zlomek RTP paketů uvnitř shlukové periody od začátku relace, které byly buď ztraceny, nebo vyřazeny během přenosu. Shluková perioda je období, při němž bylo ztraceno větší množství paketů.
- **Hustota mezer** – zlomek RTP paketů mimo shlukové periody od začátku relace, které byly ztraceny nebo vyřazeny během přenosu. Mezera mezi shluky je období, při němž jsou ztráty paketů ojedinělé.
- **Trvání shluku** – střední doba trvání shlukových period od začátku relace vyjádřená v milisekundách.
- **Trvání mezery** – střední doba trvání mezer mezi shluky od začátku relace vyjádřená v milisekundách.
- **Zpoždění tam a zpět** – pole obsahující poslední údaj o cestě tam a zpět v milisekundách mezi RTP rozhraními. Tato hodnota může být odvozena například z RTCP z pole timestamp.
- **Zpoždění konec-konec** – hodnota definována jako suma všech příspěvků zpoždění mezi uživateli a obsahuje také veškeré příspěvky zpoždění způsobené kódováním, přenosem a dekódováním signálu, včetně hodnoty pro dejitter buffer.
- **Úroveň signálu** – relativní úroveň signálu v poměru k 0 dBm0 referenci vyjádřena v decibelech.
- **Úroveň šumu** – hodnota opět vyjádřena v decibelech a získána jako odstup úrovně signálu během ticha (neaktivity) k referenční úrovni 0 dBm0.
- **RERL** – vyjadřuje hodnotu ztrát vracejícího se zbytkového echa. Je dána jako součet ERL (Echo return loss) a ERLE (Enhancement Echo return loss). Jde tedy o ztrátu vracejícího echa, které se v případě použití echo kancelátoru připočítává ERLE. Jednotkou je decibel.
- **Gmin** – udává hraniční úroveň, při které považováno, že aktuální perioda je zrovna mezera (gap).
- **R-faktor** – parametr udávající kvalitu spojení. Dosahuje hodnot v rozmezí 0-100, kde vyšší hodnota znamená vyšší kvalitu. Tato hodnota je získána jako výstup z výpočetního modelu dle doporučení ITU-T G.107 [3].

- **Externí R-faktor** – hodnota udávající R-faktor získaný během přenosu mimo IP síť. Volání může být inicializováno do nebo z například technologie GSM a toto pole obsahuje příspěvek R-faktoru z této celulární sítě.
- **MOS-LQ a MOS-CQ** – parametry vyjadřující odhadovanou kvalitu spojení vztaženou k subjektivnímu ohodnocení. Přesnějšimu popisu těchto hodnot jsou věnovány pozdější kapitoly.
- **Konfig. parametry** – obsahuje pole jako PLC, JBA a JB rate. V podstatě se jedná o mechanismy maskování ztrátovosti a variability zpoždění.
- **Parametry jitter bufferu** – obsahují aktuální hodnoty parametrů popisujících chování jitter bufferu, jako je aktuální nominální hodnota v milisekundách, jeho maximální hodnota a absolutní maximální hodnota, kterou může adaptivní jitter buffer dosáhnout.

32 bitů			
BT = 7	Reservováno	Délka bloku = 8 (Block length)	
Identifikační synchronizačního zdroje (Synchronization source identifier - SSRC)			
Úroveň ztrát (Loss rate)	Vyřazené pak. (Discard rate)	Hustota shluku (Burst density)	Hustota mezer (Gap density)
Trvání shluku (Burst duration)		Trvání mezery (Gap duration)	
Zpoždění cesty tam a zpět (Round trip delay)		Zpoždění konec-konec (End system delay)	
Úroveň signálu (Signal level)	Úroveň šumu (Noise level)	RERL	Gmin
R-faktor (R-factor)	Externí R-faktor (Ext. R-factor)	MOS-LQ	MOS-CQ
Konf. param. (RX config)	Reservováno	Jitter buffer - aktuální hodnota (JB nominal)	
Jitter buffer - max. zpoždění (JB maximum)		Jitter buffer - absolutní max. zpoždění (JB abs max)	

Obr. 2.4 Struktura RTCP XR – VoIP metrics datagramu

2.2 SIP protokol

SIP (Session Initiation Protocol) je od roku 1996 vyvíjen organizací IETF (Internet Engineering Task Force). SIP je textový protokol, který svou strukturou připomíná například SMTP (Simple Mail Transfer Protocol) či HTTP protokoly. Jeho textová podoba napomáhá k jednoduchému

ladění, ale také k jeho rozšiřování. To je patrné také tím, že první uvolněný standard RFC 3261 [21] z roku 2002 obsahoval pouze šest základních metod a dnes jsou jeho rozšíření popsána ve více než osmdesáti dalších RFC, které se SIPem souvisejí.

SIP je řídicí protokol pracující na aplikační vrstvě OSI modelu a je užíván pro sestavení, modifikaci a ukončení spojení s jedním nebo více účastníky. SIP však není jediný protokol, který je k těmto úkonům potřebný a bývá doplněn o protokol další, tzv. SDP (Session Description Protocol), který je využíván k popisu vlastností účastníků spojení. Popis přenášený v tomto doplňujícím protokolu je pak využit k vyjednání parametrů spojení všech zařízení podílejících se na komunikaci.

SIP je protokol založený na bázi klient-server, což znamená, že komunikace probíhá výměnou žádostí a odpovědí. Klientem může být označováno kterékoliv koncové zařízení, jako např. IP telefon a jako server samotný aplikační server zprostředkovávající služby. Rozsah služeb SIP protokolu v dnešní době nezasahuje pouze do samotné telefonie, nýbrž může být využit jako signalizační protokol k řízení online her či Instant Messagingu [2], [13], [21].

2.2.1 Žádosti SIP protokolu

Žádosti neboli metody jsou obvykle užívány k inicializaci určité procedury, jako je sestavení, aktualizace nebo ukončení relace. Tato podkapitola obsahuje šest základních SIP žádostí definovaných v již dříve zmíněném doporučení RFC 3261 [2], [21].

- **INVITE** – žádost sloužící k inicializaci spojení nového nebo k záměně parametrů spojení stávajícího.
- **ACK** – metoda potvrzující přijetí konečné odpovědi na žádost INVITE. Sestavení spojení využívá tzv. „3-way hand-shake“. Podstatou je, že volaný periodicky opakuje odpověď (např. 200 OK), dokud nepřijme ACK od iniciátora spojení, čímž je indikováno, že odpověď byla doručena správně.
- **BYE** – metoda sloužící k ukončení stávajícího spojení.
- **CANCEL** – slouží k ukončení teprve sestavovaného spojení. Smí být použito, pokud volaný ještě nepotvrdil žádost INVITE konečnou odpovědí a volající chce sestavované spojení ukončit.
- **REGISTER** – žádost o registraci nebo odstranění registrace uživatele. Dojde ke svázání logické adresy uživatele s jeho fyzickým umístěním (IP adresa a port). Registrace jsou časově limitovány a je nutno je periodicky obnovovat.

- **OPTIONS** – speciální metoda sloužící k zjištění schopností a vlastností SIP zařízení. Má obdobnou strukturu jako žádost INVITE, ale nesestavuje spojení, pouze přijme odpověď.

2.2.2 Odpovědi SIP protokolu

Každá žádost uvedená v kapitole 2.2.1, s výjimkou žádosti ACK, která je v podstatě potvrzení žádosti INVITE musí být zodpovězena. Odpovědi mají velmi podobnou strukturu jako SIP žádosti, s rozdílem v prvním řádku. Typ odpovědi je dán jejím kódem, který je vyjádřen celým číslem z rozsahu 100 až 699 [2], [13], [21]. Dle prvního čísla kódu lze odpovědi rozdělit do šesti tříd:

- **1xx** – dočasné informativní odpovědi na přijaté žádosti, aby se zabránilo jejich znovu posílání. Výsledek zpracování odpovědi ještě není znám. Typickým příkladem odpovědi tohoto typu jsou zprávy jako 100 (Trying) nebo 180 (Ringing).
- **2xx** – jedná se o pozitivní finální odpovědi. Požadavek byl akceptován a úspěšně zpracován. Nejběžnějším příkladem odpovědi tohoto typu je zpráva 200 OK.
- **3xx** - odpovědi užívané k přesměrování. Podávají informace o novém umístění účastníka služby. Po přijetí takovéto odpovědi je třeba vytvořit požadavek nový dle informací obsažených v této odpovědi.
- **4xx** – jedná se o opak 2xx odpovědí, jde tedy o konečné negativní odpovědi a znamenají problém na straně odesílatele a žádost tedy nemohla být zpracována.
- **5xx** – jedná se o zprávy indikující o problémech na straně serveru. Žádost je s největší pravděpodobností v pořádku, avšak server nebyl schopen ji správně zpracovat a tak je požadavek většinou odesílán znovu.
- **6xx** – indikuje globální chybu a požadavek nemůže být zpracován na žádném serveru. Typickým příkladem je zpráva 603 (Decline response) oznamující odmítnutí žádosti o sestavení spojení.

2.2.3 Struktura SIP hlaviček

Struktura SIP hlaviček je vždy v několika málo parametrech odlišná v závislosti na aktuálním požadavku či odpovědi [2]. Parametry zde vybrány a popsány jsou nejpoužívanější:

- **Request URI**- obsahuje informaci o typu zprávy (např. INVITE), dále request URI, což je adresa dalšího skoku, podle které směřujeme.
- **Via** – obsahuje adresu klienta, který žádost posílá nebo adresy všech SIP serverů po cestě, přes které požadavek prošel a přes které se bude vracet odpověď.
- **From** – identifikuje iniciátora spojení pomocí jeho URI adresy.

- **To** – identifikuje příjemce spojení pomocí jeho URI adresy.
- **Contact** – obsahuje skutečnou IP adresu a port, na které odesílatel očekává další žádosti odesílané volaným účastníkem. Využívá se, pokud není nutno žádost posílat přes SIP proxy, ale může být poslána napřímo, např. při ukončení spojení a odesílání požadavku BYE.
- **Call-ID** – jedná se o unikátní identifikátor dialogu, který má za úkol identifikovat všechny zprávy náležící jednomu volání. Všechny tyto zprávy mají právě stejný obsah pole Call-ID.
- **CSeq** – číslo žádosti v rámci dialogu. Jedno spojení obvykle obsahuje několik dialogů a díky tomuto poli lze jednoduše opakovat přenos některých zpráv, dojde-li k jejich nespolehlivému přenosu v důsledku UDP protokolu.

Za hlavičkou žádosti se obvykle nachází tělo zprávy, které je odděleno prázdným řádkem. Např. u zprávy INVITE tělo žádosti obsahuje popis médií vyhovujících iniciátorovy spojení. Toto tělo zprávy bývá označováno jako již bylo dříve zmíněno SDP. Ukázka celé SIP zprávy obsahující žádost INVITE je zobrazena na obrázku 2.5.

```
Session Initiation Protocol
Request-Line: INVITE sip:1194@158.196.146.12;transport=udp SIP/2.0
Method: INVITE
Request-URI: sip:1194@158.196.146.12;transport=udp
[Resent Packet: False]
Message Header
Via: SIP/2.0/UDP 192.168.1.23:64196;branch=z9hG4bK-d8754z-5d28fead3f81b044-1---d8754z-;rport
Max-Forwards: 70
Contact: <sip:7002@93.185.0.178:64196;transport=udp>
To: <sip:1194@158.196.146.12>
From: "7002"<sip:7002@158.196.146.12>;tag=4dfe8901
Call-ID: MzkyNDMxZjcxDYzMjAxMzg5MzY3NDI3YTIwZWQwMGE.
CSeq: 2 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
Content-Type: application/sdp
User-Agent: X-Lite Beta release 4.0 Beta stamp 54293
Content-Length: 411
Message Body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): - 8 2 IN IP4 192.168.1.23
Session Name (s): CounterPath Bria
Connection Information (c): IN IP4 192.168.1.23
Time Description, active time (t): 0 0
Media Description, name and address (m): audio 31054 RTP/AVP 0 98 8 3 101
Media Attribute (a): sendrecv
Media Attribute (a): rtpmap:98 iLBC/8000
Media Attribute (a): rtpmap:101 telephone-event/8000
Media Attribute (a): fmtp:101 0-15
Media Attribute (a): alt:1 3 : gAva1Cpp +5ECqfK6 192.168.170.1 31054
Media Attribute (a): alt:2 2 : +RD/en3m 8H6ag1md 192.168.63.1 31054
Media Attribute (a): alt:3 1 : 3aAavVsN 9DPGgkP3 192.168.1.23 31054
Media Attribute (a): x-rtp-session-id:0AF0B4E51B674D8CACE74CC81309EB0C
```

Obr. 2.5 Ukázka SIP žádosti INVITE

3 Faktory ovlivňující kvalitu řeči v IP

VoIP technologie je levnější variantou oproti klasické telefonii založené na přepínání okruhů. Toho je dosaženo přenosem všech užitečných dat, od signalizace po samotný digitalizovaný hlas, přes různé paketově orientované sítě, které jsou využívány dalšími technologiemi a systémy, tudíž veškerá kapacita přenosových médií není dedikována čistě pro samotnou telefonii. Je tedy zřejmé, že při hovoru se samotný přenos hlasu potýká s různými překážkami, které se ve výsledku podepíší na jeho kvalitě, z pohledu koncového uživatele. Faktory, které mají hlavní podíl na výsledné kvalitě hlasu při jeho přenosu přes IP síť, jsou ztrátovost paketů (packet loss), zpoždění (delay), variabilita zpoždění (jitter), echo a samotná vhodnost výběru hlasových kodeků. Tyto jednotlivé vlivy budou podrobněji popsány v následujících podkapitolách [1], [2], [18].

3.1 Ztrátovost paketů

Ztrátovost paketů nesoucích užitečná hovorová data může být zapříčiněna několika důvody jako:

- nedostatečná kapacita linky (či linek),
- přílišné zpoždění paketu na cestě – paket dorazil k cíli příliš pozdě
- nebo ztrátovost způsobená výpadky v síti.

Architekt navrhující VoIP systém se musí ujistit, že přenosová IP síť má dostatečnou kapacitu k obslužení požadovaného počtu účastníků a nebude docházet k jejímu přetěžování, tedy síťové zařízení nebudou nucena preventivně zahazovat náhodně vybrané pakety či pakety držet v obslužných frontách a nedojde tedy k degradaci kvality obslužených hovorů z důsledku ztrátovosti paketů. V některých velkých korporátních případech je dokonce nutné vyčlenit vlastní LAN síť dedikovanou pouze na VoIP a ostatní data jsou tedy obsluhována na síti vlastní [18].

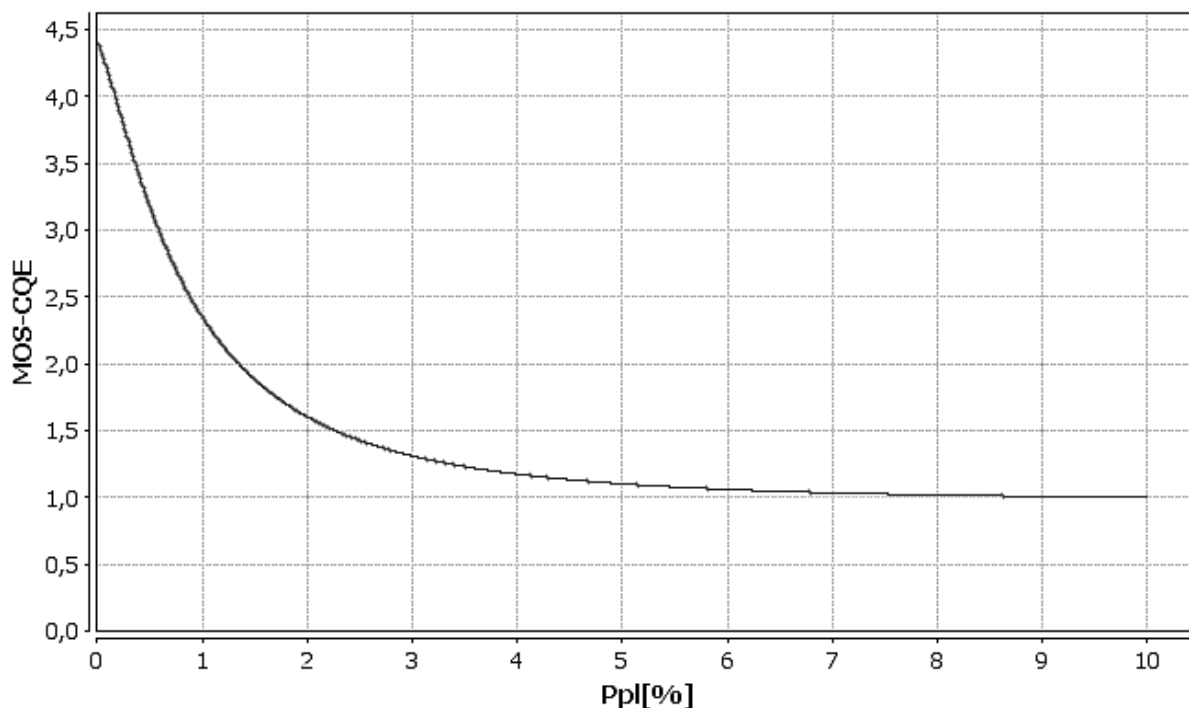
Jako jednoduchý příklad si představme síť o přenosové kapacitě 100 Mbit/s. Každý procházející RTP paket bude obsahovat 30 ms řečového vzorku, tedy 24 bytů dat. Hlavičky a bezpečnostní prvky RTP protokolu a všech protokolů ostatních, jako je UDP, IP a Ethernet přidá dalších 78 bytů zátěže, což nám dává ve výsledku 102 bytů velký Ethernetový rámec, který se opakuje každých 30 ms, tedy rozestup činí 30 ms. Za ideálních podmínek bychom tedy na takovéto 100 Mbit/s lince byli schopni obslužit omezený počet kanálů a to 3676. Tento výsledek vychází z následující rovnice:

$$Ch = \frac{BW}{P_S} \cdot \Delta t_s \quad [-; bit/s, bit, s] \quad (3.1)$$

Počet kanálů Ch lze tedy vyjádřit jako poměr kapacity přenosové rychlosti v bitech za sekundu k velikosti zátěže P_S v bitech vynásobený rozestupem mezi pakety se vzorky [2], [18]. Tedy:

$$Ch = \frac{BW}{P_S} \cdot \Delta t_s = \frac{100 \cdot 10^6}{102 \cdot 8} \cdot 30 \cdot 10^{-3} \approx 3676 \quad (3.2)$$

Ztracené pakety nejsou zdrojem znovu odesílány, neboť u hovoru je přirozená interaktivní komunikace a při každém čekání na replikaci ztraceného paketu a pozastavení paketů dalších v nějaké frontě by mělo za následek narušení této interaktivnosti. Ztrátovost se tedy v hlase projeví jako zapraskání či drobná mezera, kterou je však náš mozek do jisté míry schopen zamaskovat. Dostatečné, srozumitelné kvality hovoru lze dosáhnout do ztrátovosti přibližně 2 až 3 %, od této hodnoty se již ztrácí význam jednotlivých slov a kontext přestává být srozumitelný. Na obrázku č. 3.1 je pro ilustraci zobrazena závislost kvality v jednotkách MOS na velikosti okamžité ztrátovosti pro kodek G.711 A-law.



Obr. 3.1 Závislost kvality řeči na ztrátovosti paketů (G.711 A-law)

Ovlivnění kvality řeči ztrátovostí paketů je ovšem také dáno charakterem jeho rozložení. Je jasné, že posluchačův subjektivní názor bude odlišný při ztrátovosti paketů o velikosti 1% při shlukovém nebo čistě náhodném charakteru. Ke shlukovému rozložení ztrát může docházet např. při

dočasných přerušeních či poruchách na přenosové cestě či funkčních poruchách síťových zařízení, kdežto náhodný charakter je dán často právě špatnou volbou obsluhy paketových front.

Existuje několik způsobů, jak lze aplikovat QoS na IP síť, aby byl zajištěn bezproblémový přenos hlasového obsahu přes tuto síť. V dnešní době existuje vícero metod speciálního značkování hlasových dat, dle kterých jsou pak různé metody obsluhy paketových front schopny upřednostnit nebo naopak pozdržet určitý obsah či mu dedikovat určitou minimální šířku přenosové kapacity [2], [18].

3.2 Zpoždění a variabilita zpoždění

Celkové zpoždění v paketově orientovaných sítích je opět způsobeno vícero vlivy a je složeno z několika složek. Nejzákladnější a neoddělitelnou složkou zpoždění je doba samotného přenosu fyzického signálu po médiu, a zpoždění vzniklá ve všech síťových zařízeních po cestě. Čím delší trasu jsou pakety nesoucí hlas nuceny urazit (např. mezi kontinenty), tím je toto zpoždění větší. Jak v případě optických, tak i v případě metalických vedení lze však rychlost šíření signálu téměř přirovnat k rychlosti světla. V případě síťových prvků záleží na typu zařízení, tedy zdali jde o např. router, switch nebo firewall, protože každé zařízení potřebuje zkontrolovat údaje z hlaviček odlišné vrstvy a tedy čím hlubší zkoumání, tím delší potřebný čas na všechnu režii [18].

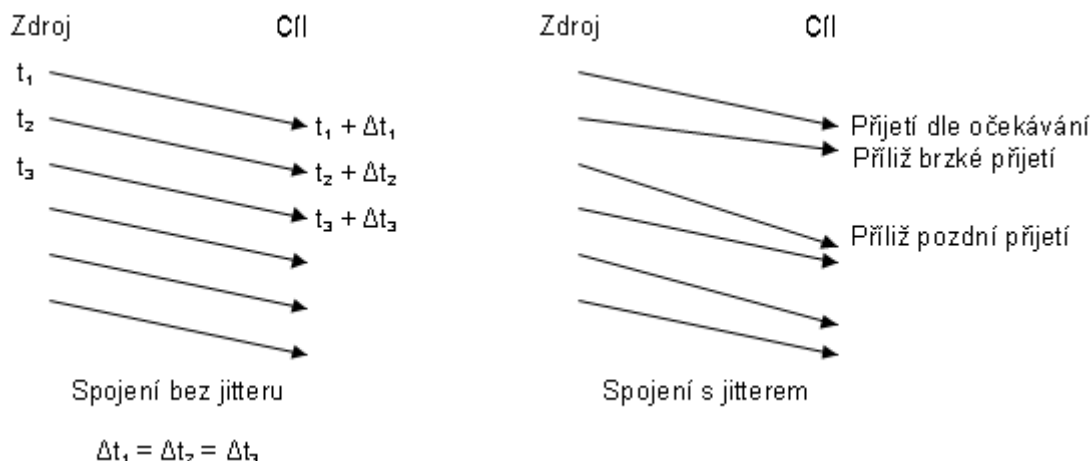
Další, z části deterministický prvek zpoždění je závislý na samotném výběru hlasového kodeku. Každý hlasový kodek používá mírně odlišný kompresní mechanismus, aby byl schopen zlepšit parametr požadavku nároku na šířku pásma. Je jisté, že kompresní mechanismus je ztrátová operace, která se již sama o sobě podepíše na výsledné kvalitě hlasu, ale také samotné zpracování kompresním algoritmem vnese jisté zpoždění.

Celková maximální doba zpoždění u VoIP hovoru, tak aby jeho obsah byl zřetelný, a rozeznatelný se pohybuje v rozmezí 150 až 500 ms, avšak v reálném provozu se nedoporučuje nasazení IP telefonie v případě, že zpoždění je konstantě vyšší než 250 ms.

Dalším faktorem negativně ovlivňujícím kvalitu hovoru je variabilita zpoždění, známá také jako jitter. Jedná se o rozptýl zpoždění, který má za následek rozdíl mezi očekávanou a skutečnou dobou příjmu paketu. Ve své podstatě mohou nastat pouze tři varianty – paket přijde dle očekávání a na čas, paket přijde příliš brzy nebo naopak příliš pozdě. Ukázka vlivu jitteru na odstup mezi pakety je znázorněna na obrázku č 3.2 [2].

Při přijímání RTP paketů nesoucích hlasové data se většinou uplatňuje tzv. jitter buffer, jehož úkolem je vliv jitteru co nejvíce minimalizovat, tak aby byly zachovány podmínky interaktivity

hovoru (je tedy jasné, že buffer nemůže pakety držet po dobu několika sekund). Jitter buffer je ve své podstatě jakási vyrovnávací paměť, ze které jsou posléze data předávány k dekódování. Tuto mezipaměť je vhodné nastavit na násobky dob trvání kódovaných hlasových fragmentů vzniklých při paketizaci. Některá zařízení vybavena sofistikovanými algoritmy pro řízení jitter bufferu dokážou tuto dobu řídit dle aktuální potřeby a stavu [2], [18].



Obr. 3.2 Jitter a jeho vliv na rozestup mezi pakety

3.3 Echo

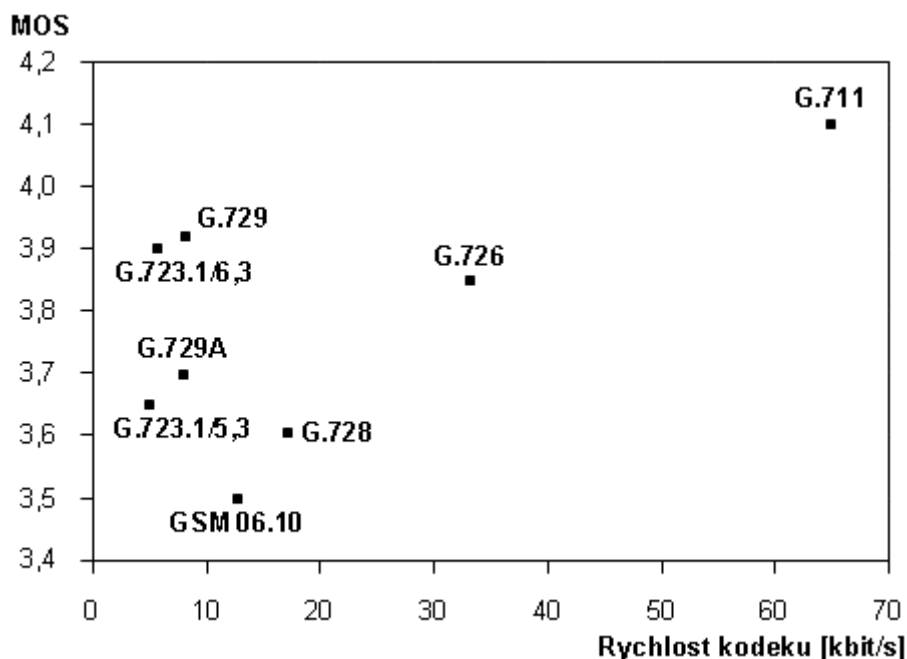
Echo u „čistých“ VoIP hovorů, tedy pouze mezi IP účastníky, kde k přenosu není nutná žádná stávající analogová či digitální telefonní síť je téměř neprojevující se faktor. Avšak uživatelé VoIP neočekávají omezení spojování hovorů čistě mezi IP účastníky, ale i mezi účastníky využívající technologie jako ISDN či GSM. Z tohoto důvodu musí na trase docházet ke konverzím mezi IP a konvenční telefonní sítí nebo také ke konverzím mezi digitálním a analogovým signálem, což má za následek vznik echa, tedy ozvěn.

V zásadě rozlišujeme dva základní druhy echa. Jedná se o hybridní echo, které vzniká především na místech konverze mezi analogovým a digitálním signálem z důvodu impedančních nevyvážeností. Druhým typem echa je akustické, které vzniká tak, že mikrofon zachytí obsah vycházející ze sluchátka, což má za následek „rozpískání“ hovoru. Většina dnešních přístrojů však nabízí již zabudované sofistikované ať už softwarové nebo hardwarové prostředky k potlačení tohoto echa. Není neobvyklé použití reproduktorů místo telefonního sluchátka, tedy potlačení tohoto echa je nutností [18].

3.4 Výběr kodeku

Kodek je mechanismus, který slouží ke zmenšení jinak zbytečně velkého objemu audiovizuálních dat. Slovo kodek vzniklo složením slov kodér a dekodér, což jsou zařízení, které jsou schopna data na jedné straně zakódovat a na straně druhé dekodovat. Při použití počítače je typickým vstupem kodeku zvuková karta, která data získaná ze vstupu mikrofону zpracuje – zakóduje – a předá je RTP vrstvě. Na druhé straně RTP vrstva přijatá zakódovaná data předá dekodéru, který je převede do původní podoby a pošle na výstup zvukové karty.

Kodeky velmi často používají ztrátovou kompresi, a proto dekódovaná data nejsou totožná s daty, která byla zakódována, což se samozřejmě podepíše na reprodukováné kvalitě zvuku na straně dekodéru. Teoretické maximální hodnoty MOS (parametr vysvětlen dále) pro vybrané kodeky v závislosti na bitové rychlosti kodeku jsou zobrazeny na obrázku č. 3.3. Z obrázku je patrné, že čím nižší přenosová rychlost, tedy větší komprese, se výrazně podepisuje na kvalitě řeči [2], [12], [16].



Obr. 3.3 MOS v závislosti na bitové rychlosti kodeků

4 Metodiky posuzování kvality řeči

Techniky k posuzování kvality hlasu či řeči prošly od svého počátku značným vývojem. Tento vývojový trend však nebyl vždy dán pouze technickým vývojem samotné telefonie, ale také tlakem trhu, tedy aby tato měření byla co nejvíce cenově dostupná a byla použitelná v reálných, ne pouze laboratorních, podmínkách.

V první řadě je dobré si uvést, v jakých jednotkách měřený parametr kvality řeči udáváme. Pro tato hodnocení se využívá stupnice MOS (Mean Opinion Score) (viz Obrázek č. 4.1), která je popsána v původním znění v doporučení ITU-T P.800 [7]. V průběhu času došlo k postupným úpravám a rozšířením této stupnice, aby lépe reflektovala výsledky získané z různých druhů metod. V roce 2003 proto ITU-T vydala aktualizované vydání původního doporučení v doporučení P.800.1 [8], jež se používá dodnes.

Metody k posuzování kvality hovoru lze prakticky rozdělit do dvou základních skupin dle hloubky hodnocení a to na:

- konverzační testy,
- poslechové testy.

Hlavní rozdíl těchto skupin vyplývá již z jejich názvu. Konverzační testy jsou založeny na vzájemné interakci dvou subjektů, a tedy tyto testy zohledňují jisté parametry celého přenosového řetězce, které se v čistě poslechových testech nemají šanci promítnout, jako je např. zpoždění. Je tedy zřejmé, že s konverzačními testy jsme ve výsledku schopni dosáhnout realističtějších výsledků, avšak za cenu větší náročnosti celého testovacího systému. Z tohoto důvodu jsou mnohem častěji využívány testy poslechové, které nedosahují takové věrohodnosti, jako testy konverzační, ale možnost jejich použití je mnohem snazší. Je dobré také zmínit, že samotný parametr kvality hlasu lze tedy rozdělit, dle testu, kterým byla získán, na konverzační (CQ – Conversational Quality) a poslechovou (LQ – Listening Quality) [1], [12].

Dalším hlediskem, dle kterého lze dělit metodiky pro měření kvality hlasu je samotný přístup k získávání výsledků, na:

- subjektivní metody,
- objektivní metody.

Podrobnějšímu popisu těchto dvou základních skupin jsou věnovány další podkapitoly této kapitoly. V tabulce č. 4.1 je pro ilustraci uvedeno rozdělení druhů MOS stupnice v závislosti na testech či metodách měření [1], [7], [8].



Obr. 4.1 Poslechová MOS stupnice dle ITU-T

Tabulka 4.1: Rozdělení MOS stupnic dle ITU-T P.800.1

Metoda měření	Poslechový typ	Konverzační typ
Subjektivní	MOS-LQS	MOS-CQS
Objektivní	MOS-LQO	MOS-CQO
Odhadovaná	MOS-LQE	MOS-CQE

4.1 Subjektivní metody hodnocení

Subjektivní metody k posuzování kvality řeči v telefonii jsou, jak již název napovídá, založeny na hodnocení lidskými uživateli, tedy subjekty. Základní princip subjektivních metod je postaven na přehrávání hovorových vzorků dostatečně velké skupině posluchačů a je na subjektu, tedy každém účastníkovi testování, každému takovému vzorku přiřadit známku podle stupnice MOS dle svého subjektivního úsudku. Takto získané výsledky je ještě nutno před závěrečnou prezentací statisticky vyhodnotit.

Subjektivní metody hodnocení mají několik druhů testů, které jsou specifikovány a standardizovány organizací ITU-T v jejich doporučeních. Mezi tyto nejvyužívanější zástupce patří ACR (Absolute Category Rating) a DCR (Degradation Category Rating). V případě ACR posluchači udělují právě jednu známku dle poslechové MOS stupnice (viz obrázek 4.1) každému přehrávanému vzorku hlasu. Známky jsou poté ode všech subjektů sečteny pro jednotlivé vzorky a výsledným parametrem kvality pro daný vzorek je jejich aritmetický průměr v jednotkách MOS (MOS-LQS). U metody DCR je posluchačům nejprve poskytnut vzorek hlasu „původní“ nebo „originální“, tedy před přenesením přes telekomunikační systém a posléze vzorky testované. Úkolem subjektů je stanovit a vyhodnotit degradaci těchto přenesených vzorků v porovnání se vzorkem původním [7], [8]. Další standardizované metody pro subjektivní hodnocení kvality řeči včetně jejich typu jsou uvedeny v tabulce 4.2.

Tabulka 4.2: Přehled subjektivních testovacích metod.

Test	Typ testu
Converstional Opinion	Konverzační
ACR	Poslechový
Quantal-Response Detectability	Poslechový
DCR	Poslechový
Comparison Category Rating	Poslechový

Hodnocení získána takovýmto způsobem jsou zatížena určitou chybou, která je způsobena různými faktory ovlivňující samotný subjekt. Jedná se např. o psychické rozpoložení, soustředěnost, ochota nebo zkušenosti posluchače, proto při hodnocení stejného vzorku řeči je možné dosáhnout určitého intervalu hodnot. Pro hodnocení takovýmto způsobem je tedy nutno zajistit co možná největší počet subjektů, aby získané výsledky v jejich výsledné interpretaci co nejvíce odpovídaly skutečnosti. To ovšem znamená obrovské časové a finanční náklady, které zabraňují použití takovýchto metod pro testování přenosových systémů v reálném čase. Z tohoto důvodu je doporučeno a téměř výhradně používáno metod objektivních [11], [12].

4.2 Objektivní metody hodnocení

Použití objektivních metod k hodnocení kvality řeči v telefonii odstraňuje nutnost použití živých subjektů nahrazením vhodným matematickým modelem či algoritmem. Výstupem takovýchto metod je opět hodnota MOS (dle tabulky č. 4.1) nebo hodnota jiná, která je však podle určité matematické relace na hodnotu MOS převeditelná. Cílem objektivních metod hodnocení je co možná nejpresněji odhadnout nebo předpovědět hodnotu MOS, která by byla získána subjektivními metodami za použití dostatečného množství hodnotících subjektů. Kvalita a přesnost objektivního modelu je vždy dána korelací výsledků získaných touto metodou a subjektivního testu.

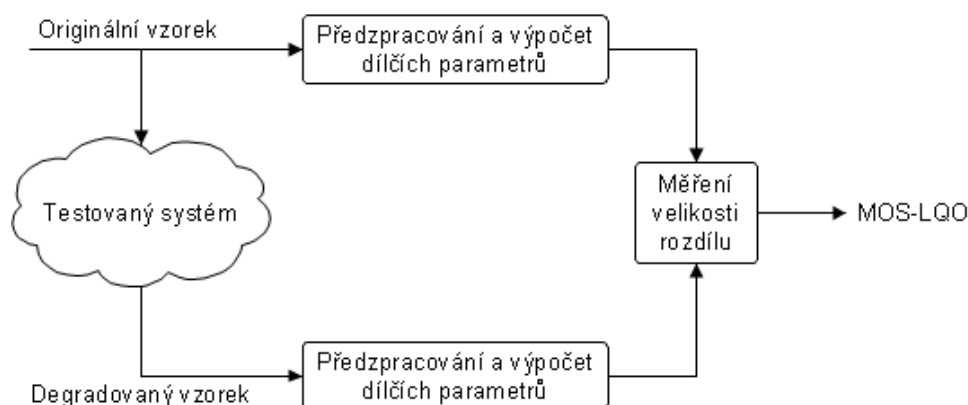
Objektivní metody se dále dělí do dvou skupin na:

- intrusivní,
- neintrusivní.

Intrusivní měření kvality je založeno na porovnávání původního (originálního či referenčního) vzorku řeči se vzorkem degradovaným, tedy po průchodu přenosovým řetězcem, za použití vhodného algoritmu. Neintrusivní metody naopak používají ke svým výpočtům a stanovení kvality pouze vzorky degradované nebo dokonce dokážou kvalitu odhadovat pouze z parametrů přenosové cesty [1], [11].

4.2.1 Intrusivní testy

Podstatou všech intrusivních (nebo také input-to-output) testů je porovnávání původního a přeneseného vzorku hlasu. Výsledkem takovýchto testů je dle doporučení P.800.1 [8] hodnota označovaná jako MOS-LQO (viz tabulka č. 4.1). Obecná architektura intrusivních měření je zobrazena na obrázku č. 4.2. První funkční blok má na starosti předzpracování vstupujícího zvukového signálu, jak původního, tak degradovaného, a extrakci všech důležitých parametrů a informací z těchto vzorků. Zde jsou také tyto vzorky transformovány do určité domény podle způsobu a potřeby testování.



Obr. 4.2 Obecná architektura intruivních testů

Těmito doménami jsou:

- časová,
- spektrální,
- smyslová.

Druhý funkční blok (měření velikosti rozdílu) zahrnuje měření nesourodosti původního a přeneseného vzorku na základě zvolené domény [11], [12].

4.2.1.1 Měření na základě časové domény

Měření na základě časové domény jsou převážně uplatňována v systémech používajících analogové kódování, nebo-li kódování průběhu a tvaru vlny vstupního signálu, kde cílové zařízení má za úkol reprodukovat přenesenou vlnu pomocí zakódovaných parametrů. Typicky měřené parametry u takových testů jsou hodnoty odstupe úrovně signálu od úrovně šumu SNR (Signal to Noise Ratio). Při tomto měření je signálem myšlena užitečná informace přenášená přes komunikační médium a šumem všechny signály ostatní a signály rušící. Tvary a průběhy vln původního a přeneseného signálu jsou porovnávány přímo, tudíž zde hraje velkou a rozhodující roli časová synchronizace těchto signálů.

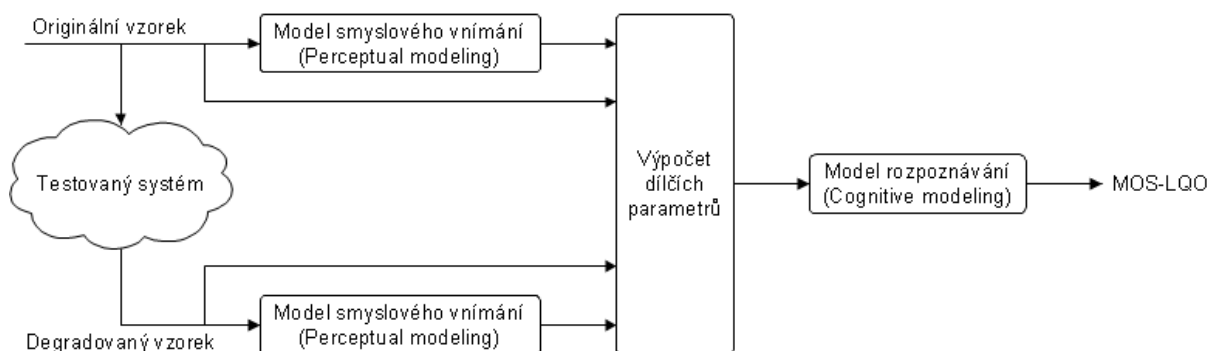
Pokud nedosáhneme takové správné časové synchronizace, výsledky těchto testů jsou často velmi zkreslené a odlišné od skutečnosti [11].

4.2.1.2 Měření na základě spektrální domény

Objektivní měření na základě spektrální domény jsou o něco věrohodnější než metody založené na doméně časové, neboť jsou více odolné vůči nesrovnalostem v časové synchronizaci a fázovém posuvu mezi původním a přeneseným vzorkem signálu. Tato měření jsou silně svázána s vlastnostmi použitých hlasových kodeků a modelů reprodukce hlasu. Schopnost efektivně popsat posluchačovu sluchovou reakci je limitována právě omezením řečových reprodukčních modelů [11].

4.2.1.3 Měření na základě smyslové domény

Objektivní metody založené na této doméně jsou nejrozšířenější a nejvěrohodnější pro měření kvality řeči. Na rozdíl od měření na základě spektrální domény nejsou závislé na použitých modelech reprodukce hlasu, ale jsou založeny na lidském sluchovém vnímání a z toho důvodu mají ve srovnání se subjektivními metodami největší potenciál a nejlepší výsledky. Při těchto měřeních je zvukový signál transformován do modelu založeného na lidských smyslech za použití konceptů psychofyzického sluchového vnímání. Takovéto koncepty jsou např. kritické pásmo spektrálního rozlišení, kmitočtová selektivita, křivka vnímané hlasitosti a další [11]. Zjednodušené blokové schéma takových měření je vyobrazeno na obrázku č. 4.3.

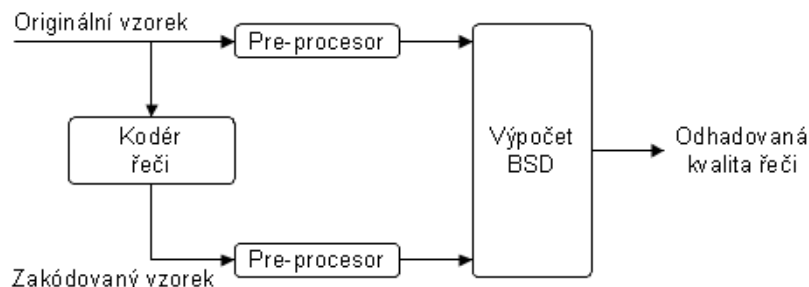


Obr. 4.3 Obecné schéma intrusivních testů na základě smyslové domény

Nejnámější a nepoužívanější zástupci objektivních intrusivních metod:

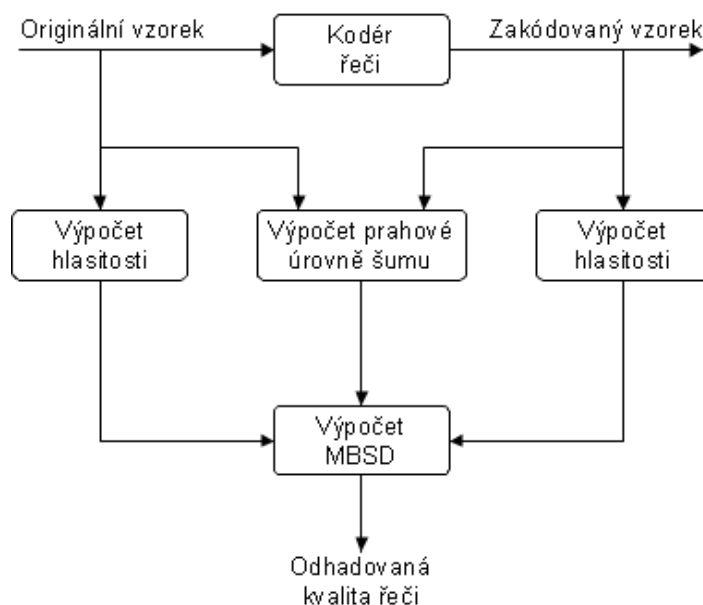
- **BSD měření (Bark Spectral Distortion)** - slouží k objektivnímu měření zkreslení na základě spočitatelných vlastností sluchového vjemu. Výsledkem BSD měření je průměr čtverce Euklidovské vzdálenosti mezi spektrálními vektory původního a zakódovaného vzorku řeči. Měření je založeno na emulování několika známých vlastností lidského ucha, jako jsou prohýbání kmitočtového spektra, změna citlivosti ucha v závislosti na změně frekvence a rozdíl mezi úrovní hlasitosti a subjektivní

stupnicí hlasitosti. Zjednodušené blokové schéma měření metodou BSD je znázorněno na obrázku č. 4.4 [11], [12].



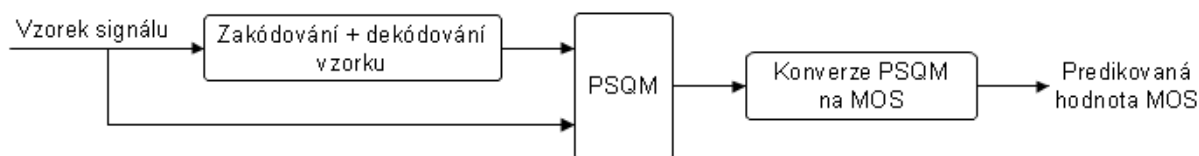
Obr. 4.4 Blokové schéma metody BSD

- **MBSD a EMBSD měření (Modified BSD, Enhanced Modified BSD)** - MBSD je modifikace klasického BSD, ve kterém je začleněn koncept prahového krytí hluku, které se liší pro slyšitelné a neslyšitelné rušící vlivy. Používá stejné maskování hluku, jako je použito u transformačního kódování zvukových signálů. Další změnou oproti původnímu BSD je způsob výpočtu výsledné kvality vzorku, kde se porovnávají očekávané hlasitosti jednotlivých vzorků (přenesený, zakódovaný). Blokové schéma měření metodou MBSD je znázorněno na obrázku č. 4.5. EMBSD je další rozšíření metody MBSD, kde byly pozměněny některé dílčí procedury, a byl zvolen nový model rozpoznávání [11], [12].



Obr. 4.5 Blokové schéma metody MBSD

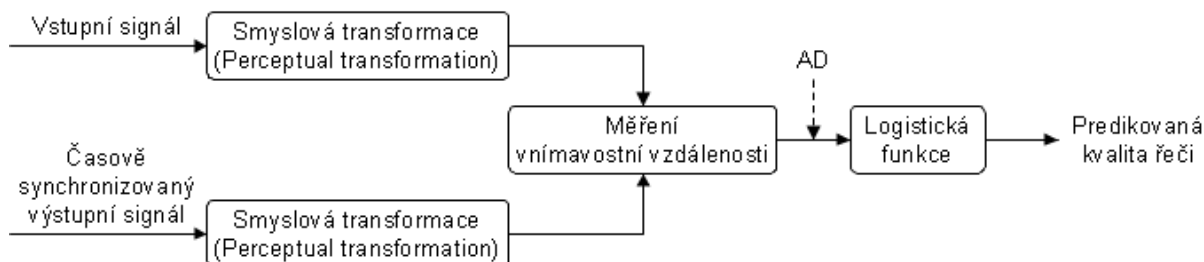
- **PSQM (Perceptual Speech Quality Measurement)** - metoda popsaná v doporučení ITU-T P.861. Jedná se o matematický proces, který poskytuje přesné objektivní měření subjektivní kvality řeči. PSQM bylo vytvořeno pro použití a aplikaci na signály v základním telefonním pásmu (tedy 300 - 3400 Hz) s použitím kodeků a vokodérů s nízkými přenosovými rychlostmi. Při měření metodou PSQM je vzorek hlasového signálu zakódován a posléze dekódován systémy, jež testovaný přenosový řetězec využívá k normálnímu provozu. Tento výsledný signál je poté časově synchronizován se vzorkem původním. Následně PSQM algoritmus porovná tyto dva signály na základě faktorů lidského vnímání, jako je citlivost na kmitočet a hlasitost. Výsledkem algoritmu je hodnota PSQM, která určuje „vnímavostní vzdálenost“ mezi původním a přeneseným vzorkem. Užitím vhodného algoritmu lze tuto hodnotu přepočítat na hodnotu dle stupnice MOS. Nevýhodou této metody je, že nedokáže úplně přesně pracovat se všemi faktory ovlivňujícími kvalitu řeči, jakým je např. ztrátovost (packet loss) a jiné časové poruchy [1], [11], [12]. Obecné blokové schéma této měřicí metody je znázorněno na obrázku č. 4.6 .



Obr. 4.6 Blokové schéma metody PSQM

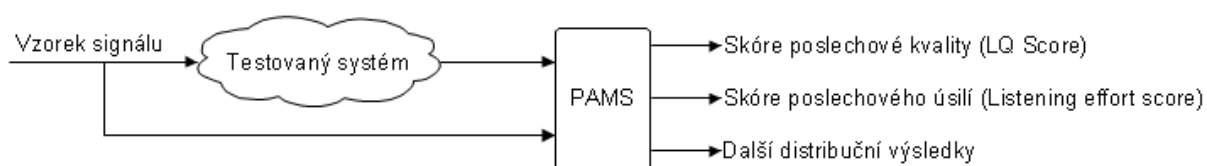
- **PSQM+ (PSQM Plus)** - vychází z metody PSQM a upravuje její nedostatky týkající se časových poruch signálu a ztrátovosti. Při měření za použití metod PSQM a PSQM+ dosáhneme stejných výsledků, ale pokud dochází k velké časové deformaci výsledného signálu nebo patrné ztrátovosti paketů, metoda PSQM+ poskytne mnohem menší hodnoty (tedy přesnější) výsledné kvality než původní metoda PSQM [11], [12].
- **MNB (Measuring Normalising Blocks)** - je součástí upraveného doporučení ITU-T P.861 (PSQM) z roku 1997 jako Annex II. Jde o alternativní techniku k PSQM, k měření „vnímavostní vzdálenosti“ (Perceptual Distance), mezi původním a přeneseným vzorkem. Při tomto způsobu měření jsou uplatněny smyslové transformace jak na přenesený, tak také na původní zvukový signál před samotným měřením „vnímavostní vzdálenosti“. Metoda MNB se dále dělí podle způsobu měření na TMNB (Time MNB) a FMNB (Frequency MNB). TMNB a FMNB jsou zkombinovány se zatěžujícími faktory, kde je výsledkem nezáporná hodnota zvaná

Audиторní vzdálenost AD (Auditory Distance). Nakonec je použita logistická funkce, která dokáže převést AD hodnoty na konečnou stupnici a umožňuje korelaci se subjektivní stupnicí MOS [11], [12]. Blokové schéma tohoto způsobu měření je znázorněno na obrázku č. 4.7.



Obr. 4.7 Blokové schéma metody MNB

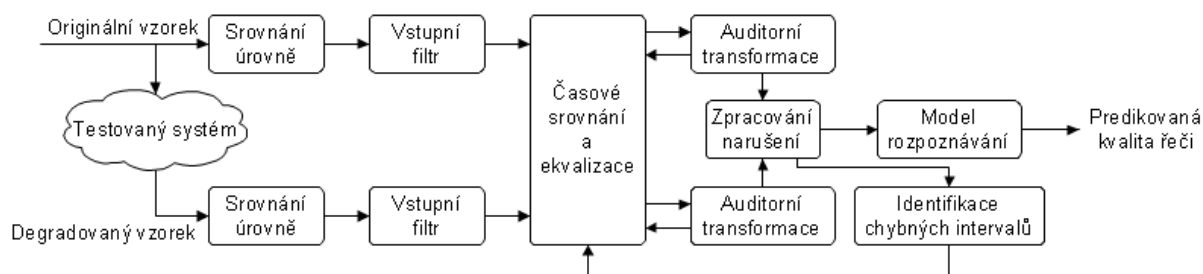
- **PAMS (Perceptual Analysis Measurement System)** - za vznikem této metody stojí společnost British Telecommunications. Metoda PAMS je v mnoha ohledech podobná metodě PSQM, avšak používá úplně odlišné techniky zpracování signálu a modely smyslového vnímání. Metoda PAMS ve svém výpočtu nezahrnuje efekt zpoždění, celkový zisk/útlum testovaného systému, časové a úrovněové zarovnání a ekvalizaci. PAMS porovnává původní a přenesený vzorek na základě časově-frekvenční domény. Toto porovnání je založeno na lidských smyslových faktorech, kde výsledkem je hodnota v rozmezí 1-5, která koreluje se škálou MOS hodnot. PAMS dále navíc vytváří tzv. Listening Effort Score, které koresponduje s ACR dle obou doporučení ITU-T P.800 [7] a ITU-T P.830. Blokové schéma tohoto způsobu měření je znázorněno na obrázku č. 4.8 [1], [11], [12].



Obr. 4.8 Blokové schéma metody PAMS

- **PESQ (Perceptual Evaluation of Speech Quality)** - jedná se o nejnovější objektivní testovací metodu. Je popsána v doporučení ITU-T. P.862 [9]. Metoda PESQ je primárně určena a vhodná pro stanovení kvality řeči u úzkopásmových telefonních signálů, ovlivněných následujícími stavy: užití kodeků tvarového a netvarového průběhu, transkódování, velikost vstupní úrovně signálu do kodéru, chyby přenosového kanálu, šum způsobený přenosovým systémem a krátké nebo dlouhé

časové poruchy. Technika výpočtu metody PESQ kombinuje robustní techniky časového zarovnání metody PAMS a přesný model smyslového vnímání metody PSQM. Samotný výpočet lze rozdělit do několika kroků. V prvním kroku model zarovná původní a degradovaný signál na stejnou konstantní výkonovou úroveň, která koresponduje s normální poslechovou úrovní používanou u testů subjektivních. Ve druhém kroku oba signály prochází vstupním filtrem, který je založen na rychlé Fourierově transformaci. V krocích dalších jsou signály časově zarovnány a „auditorně“ transformovány (Auditory transform) podobně jako u systémů PAMS a PSQM. Auditorní transformace u systému PESQ je založena na psychoakustickém modelu, který mapuje signály do přijímané hlasitosti v čase a frekvenci napodobováním určitých klíčových vlastností lidského ucha a odstraněním částí, které jsou nadbytečné a posluchačem neslyšitelné [1], [9], [11], [12]. Blokové schéma této měřicí metody je znázorněno na obrázku č. 4.9.



Obr. 4.9 Blokové schéma metody PESQ

4.2.2 Neintrusivní testy

Všechny metody uvedené v kapitole 4.2.1 byly založeny na input-to-output měření, kde vyhodnocování kvality je prováděno na základě znalostí nejen vzorku přeneseného přes komunikační řetězec, ale také vzorku původního. Avšak tato měření nesou také některá svá úskalí, jako je např. nutnost precizní synchronizace při porovnávání těchto dvou vzorků nebo přímo nutnost mít k dispozici vzorek původní [1], [12].

Intrusivní metody měření kvality řeči jsou více precizní a přesné v určení výsledné kvality, ale většinou jsou pro měření v živých sítích a pro nasazení v reálném čase nepoužitelné. Naopak neintrusivní metody pro měření kvality řeči potřebují ke svému výpočtu pouze vzorek přenesený, tedy degradovaný nebo parametry přenosového řetězce. Tyto testy mají většinou složitější výpočetní modely právě díky absenci vzorku původního. Existují dva přístupy k provádění měření užitím těchto metod:

- založeny na pravděpodobnosti (priori-based),

- založeny na znalosti (source-based).

4.2.2.1 *Přístup na základě pravděpodobnosti (priori-based)*

Tento přístup je založen na identifikaci souboru charakteristických deformací v přeneseném vzorku a naučení statistických vazeb mezi tímto konečným souborem a subjektivními názory. Většina takovýchto měření je založena na zkoumání vizuálních vlastností spektrogramů. Tyto metody vychází z práce J. Palakala a M. J. Zorana [15], kteří navrhli metodu, jak zachytit neměnné aspekty řeči ve spektrogramu užitím umělých neuronových sítí. Spektrogram je dvourozměrná grafická reprezentace spektra měnícího se v čase. Spektrogramy obsahují bohaté akustické i fonetické informace. Vyhodnocování těchto grafických reprezentací mohou být zpracovávány jak strojově, tak i lidskými odborníky s dostatečnými zkušenostmi a vědomostmi.

Další příklad metody založené na pravděpodobnosti je popsán v doporučení ITU-T P.562. Tato metoda využívá tzv. INMD (in-service nonintrusive measurement devices), které mají přístup k přenosovým kanálům a mají schopnost shromažďovat objektivní informace o probíhajících hovorech bez jejich narušení. Tato nashromážděná data jsou poté dále zpracovávána a výsledkem je předpovídaná hodnota MOS, odpovídající doporučení ITU-T P.800.1 [8].

Jednou z novějších metod pro neintrusivní vyhodnocování kvality řeči je E-model, zpracovaný v doporučení ITU-T G.107 [3]. Tato metoda s užitím INMD zařízení dokáže dosáhnout velmi přesných výsledků, ale její hlavní účel je spíše nasazení při plánování nových komunikačních systémů, než nasazení v reálném čase pro vyhodnocování aktuální kvality. Komplexnost moderních sítí vyžaduje, aby parametry přenosových cest nebyly posuzovány jednotlivě, ale byly brány v potaz všechny jejich možné kombinace a vzájemné ovlivňování. Přesně takový přístup nabízí i E-model, jehož výstupem je skalár s označením R, který se mění přímo s celkovou kvalitou hovoru. E-model je založen na metodě tzv. „equipment impairment factor“. Původní strukturu tohoto modelu vyvinul švédský expert Nils-Olof Johannesson působící ve skupině Voice Transmission Quality from Mouth to Ear, patřící pod seskupení ETSI. V letech 1997 - 2000 tento model převzala a rozpracovala studijní skupina SG12 patřící pod ITU-T a vydala jej v již zmíněném doporučení ITU-T G.107 [3].

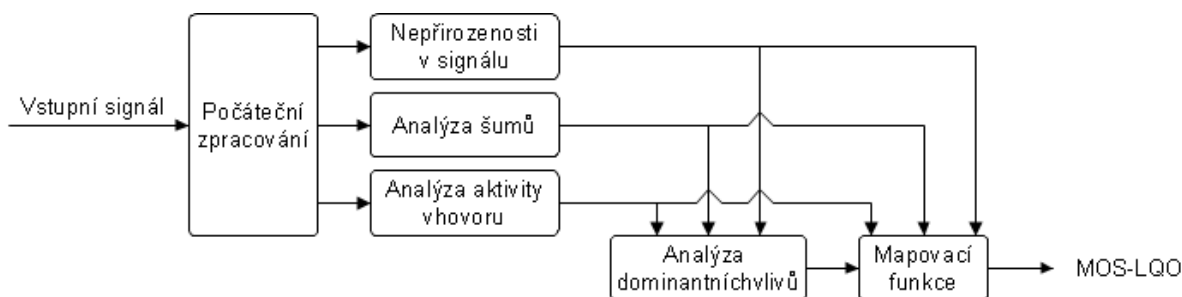
Výpočet celého modelu se skládá z několika matematických operací nad různými parametry přenosové cesty, jako jsou echa, hlasitosti šumů, zpoždění a jiné. Samotný výpočet lze rozložit do několika komponent. Rovnice pro výpočet skaláru R je definována vztahem (4.1).

$$R = R_o - I_s - I_d - I_{e_ef} + A \quad (4.1)$$

R_o představuje základní odstup signálu od šumu, ve kterém jsou zahrnuty všechny možné druhy šumu včetně šumů způsobenými elektrickými obvody zařízení a šumy způsobených na vedení.

Is zahrnuje všechny možné kombinace zhoršení, které se objevují více či méně souběžně s užitečným hlasovým signálem. Faktor Id představuje všechna zhoršení, která jsou způsobena různými kombinacemi zpoždění. Ie_ef zahrnuje zhoršení způsobené užitím určitého hlasového kodeku, projevením možné ztrátovosti paketů, a jeho odolnosti vůči ztrátovosti. V poslední řadě nám parametr A celkovou výslednou kvalitu mírně zlepšuje, neboť např. v případě satelitního telefonu je uživatel na hovor více soustředěný než při běžném použití pevného terminálu v domácím prostředí [1], [3], [4], [5], [6], [12].

V roce 2004 ITU vydala v doporučení ITU-T P.563 (single ended method for objective speech quality assessment in narrow-band telephony applications) novou metodu neintrusivního získávání ohodnocení kvality řeči. ITU tuto metodu představuje jako první neintrusivní metodu, která zahrnuje všechny možné druhy rušení a zhoršení hovorového signálu ve veřejné telefonní síti s přepojováním okruhů (PSTN). Výstupem je opět hodnota MOS korespondující s doporučením ITU-T P.800.1 [9]. Blokové schéma této metody je vyobrazeno na obrázku č. 4.10. Algoritmus analyzuje signál na základě nepřírozeností, šumů a prázdných míst (ticha) v hovoru. Z těchto analýz se vyberou aspekty, které mají na přenášený signál největší vliv. V posledním kroku je použita mapovací funkce, která pomocí matematických operací nad těmito daty vrátí hodnotu MOS-LQO [1], [12].



Obr. 4.10 Blokové schéma metody dle ITU-T P.563

4.2.2.2 Přístup na základě znalosti (source-based)

Tento přístup umožňuje univerzálnější metody než přístup na základě pravděpodobnosti výskytu zhoršení. Tyto metody mají přístup k rozsáhlému souboru všech různých typů zhoršení, kde jsou popsány porovnáním některých vlastností degradovaného signálu s modelem signálu původního.

První metody patřící do této skupiny byly založeny na tzv. PLP (perceptual-linear prediction) modelu, který porovnával „vnímavostní“ vektory extrahované z přenášeného vzorku s vnímavostními vektory odvozenými z naprosto čistých, nijak nedegradovaných vzorků. Avšak tyto metody byly příliš náročné na svůj výpočetní čas, a tudíž velmi neefektivní.

Později bylo na tuto metodu navázáno, jenom se již extrahované vektory neporovnávaly s interní databází vytvořených vektorů, nýbrž s referenčním souborem kódů, který byl z těchto vektorů vytvořen. Tento způsob samotný výpočet zrychlil, avšak výsledná kvalita je silně závislá na obsáhlosti a kvalitě tohoto srovnávacího souboru [11].

5 Návrh a vývoj aplikace pro měření kvality řeči

Praktická část této diplomové práce se zabývá návrhem a realizací jednoduchého testovacího systému sloužícího k sledování (nebo lépe řečeno k odhadování) kvality řeči na open source softwarové pobočkové ústředně Asterisk. Nasazením prostředku pro sledování kvality řeči do komunikačního prostředku, kterým je VoIP ústředna může sloužit ke spoustě věcí, jako je statistický přehled kvality sestavovaných hovorů, indikace administrátora v případě dlouhodobého rapidního poklesu kvality, podávání reportů koncovým uživatelům (zákazníkům) a jiné. Cesta, kterou jsem se při návrhu vydal, byla založena na využití čistě open source či vlastních prostředků, tedy všechny komponenty, ze kterých je výsledný produkt složen nepodléhají žádné placené licenci [14].

5.1 Specifikace vyvíjené aplikace

Jelikož je VoIP ústředna Asterisk provozována především na operačních systémech založených na platformě Linux, pro vývoj jsem zvolil uživatelsky snadno ovladatelného prostředí linuxové distribuce Ubuntu verze 10.11. Dalším požadavkem je přehledné, uživatelsky přátelské rozhraní, kterým je celá aplikace ovládána. K tomuto účelu jsem zvolil vývoj webového uživatelského rozhraní napsaného ve skriptovacím jazyce PHP s využitím Ajaxu pro spouštění řídicích skriptů.

Jako metoda k samotnému stanovování kvality řeči byl zvolen po dohodě s mým vedoucím výpočetní model - E-model [3] (viz kapitola 4). Zpracování algoritmu E-modelu jsem provedl v programovacím jazyce Java. Jelikož ale není možnost zjistit všechny parametry přenosové cesty a tedy použít rovnici dle doporučení, je použita zjednodušená forma, kde jsou některé faktory nahrazeny konstantami, které jsou dány statistickým zjištěním [3], [6]. Výsledná podoba této rovnice je následující:

$$R = 94,7688 - 1,4136 - 0 - I_{e_ef} + 0 \quad (5.1)$$

Výpočet parametru I_{e_ef} zůstal nezměněn a vychází z doporučení E-modelu – ITU-T G.107 [3]:

$$I_{e_ef} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{\frac{P_{pl}}{BurstR} + B_{pl}} \quad (5.2)$$

I_e je tabulková hodnota dle užitého kodeku, P_{pl} značí hodnotu ztrátovosti paketů udávanou v procentech, $BurstR$ je charakter ztrátovosti, tedy jedná-li se o jev náhodný či shlukový, v našem případě je hodnota nastavena pevně na 1, tedy že jde o náhodný projev ztrátovosti, neboť k analýze by bylo potřeba složitějšího systému. V poslední řadě parametr B_{pl} udává hodnotu odolnosti a schopnost maskování kodeku proti projevující se ztrátovosti paketů, což je opět tabulková hodnota vycházející

přímo z doporučení. Výstupem E-modelu, jak již bylo zmíněno, je však parametr R faktoru a pro zobrazení výsledků v jednotkách MOS je nutno na výstup E-modelu použít tzv. mapovací funkci. Pro $R < 6,5$:

$$MOS = 1 \quad (5.3)$$

Pro $6,5 \leq R \leq 100$:

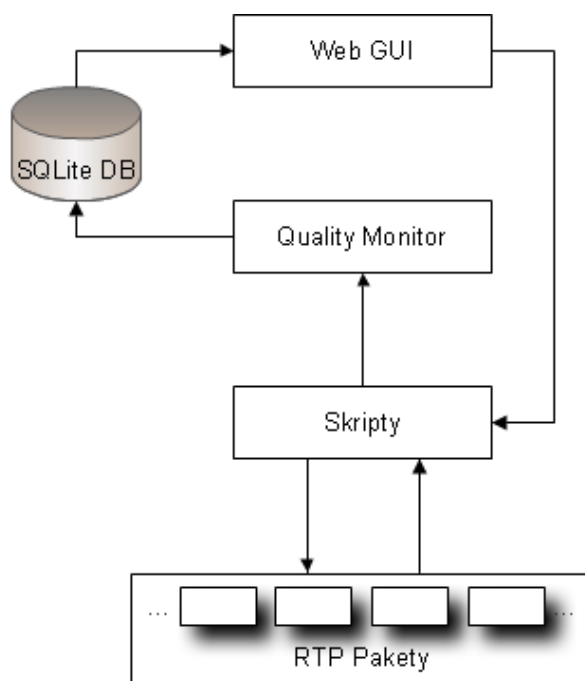
$$MOS = 1 + 0,035 \cdot R + R \cdot (R - 60) \cdot (100 - R) \cdot 7 \cdot 10^{-6} \quad (5.4)$$

A pro hodnoty $R > 100$ (u širokopásmových kodeků):

$$MOS = 4,5 \quad (5.5)$$

5.2 Struktura aplikace

Pro větší transparentnost aplikace jsem její strukturu rozčlenil do několika logických bloků, které jsou zobrazeny na obrázku č. 5.1. Tato podkapitola se bude věnovat popisu jednotlivých bloků a komponent, ze kterých je složena.



Obr. 5.1 Struktura monitorovacího systému

5.2.1 Web GUI

Webové rozhraní je hlavní uživatelskou částí celé aplikace. Skrze intuitivní ovládání pomocí tlačítek v horní části tohoto prostředí se provádí veškerá interakce s celým monitorovacím systémem.

Tento blok navrženého systému slouží také jako prostředek k zobrazení naměřených a spočtených výsledků. Struktura prezentovaných dat je následující:

Čas - Zdrojová IP - Cílová IP - Ztrátovost - MOS - R-faktor - Kodek

Pro prezentaci výsledků uvedu příklad pro několik hovorů:

16:03	192.168.21.3	192.168.21.6	0.5	61.69	3.01	G.711/G.711
16:05	192.168.21.3	192.168.21.1	0	93.36	4.41	G.711

Samotné webové rozhraní je napsáno za pomoci skriptovacího jazyka PHP s využitím Ajaxu, neboť ke spouštění či zastavení monitorovacího systému je potřeba spouštět skripty pro shell operačního systému na straně serveru. Jak již bylo dříve zmíněno, prezentovaná výsledná data jsou ve stejném formátu ukládány do SQLite 3 databáze, tak použití jazyka PHP usnadní i manipulaci s těmito daty bez použití dalších externích aplikací a konektorů. Ukázka webového rozhraní je zobrazena na obrázku č. 5.2.

Monitoring is running...						
<div>StopResultsRefreshErase</div>						
Date	From	To	PL [%]	R-factor	MOS	Codec(s)
23.04.2012 05:46	192.168.21.50	192.168.21.55	2	55.35	2.79	G.711/G.711
23.04.2012 05:55	192.168.21.50	192.168.21.55	0.5	69.6	3.38	G.711/G.711
23.04.2012 05:59	192.168.21.50	192.168.21.55	1	61.68	3.01	G.711/G.711

Obr. 5.2 Web GUI monitorovacího systému

5.2.2 Skripty

Spouštění skriptů je inicializováno pomocí webového rozhraní monitorovacího nástroje samotným zapnutím monitorování. Prakticky to znamená spuštění zachytávání síťového provozu pomocí nástroje tshark, na kterém je aktivován RTP filtr. Použití tohoto RTP filtru velmi usnadní vyhodnocování RTP streamů, neboť z těchto dat dokáže přímo získat všechny důležité parametry nutné k výpočtu pomocí zjednodušeného E-modelu (viz rovnice (5.2)). Těmito parametry jsou ztrátovost paketů, jitter, použitý kodek, zdrojová a cílová IP adresa. Příkaz v inicializačním skriptu pro zapnutí monitorování pomocí tsharku vypadá následovně:

```
tshark -a duration:900 -qp -z rtp,streams > output.log
```

Takto sesbíraná data jsou uložena do výstupu v textové podobě bohužel až po ukončení samotného tshark systémového procesu. Z tohoto důvodu a šetření místa na disku na straně serveru (sesbíraná data před vyhodnocení RTP filtrem mohou zabírat i několik GB v závislosti na délce trvání zapnutí tshark procesu a objemu síťové komunikace na serveru) je tedy nutno tento proces opakovat s určitou periodou. Za vhodnou periodu jsem zvolil 15 minut (duration:900). Jakmile jsou data získána za pomoci tsharku a RTP filtru k dispozici v textové podobě (> output.log), může být pomocí dalšího skriptu spuštěn výpočet kvality, který je hlouběji popsán v další kapitole. Po zpracování dat jsou veškeré uložené soubory z disku vymazány z důvodu šetření místa. Příkaz pro aktivaci výpočetního modelu vypadá následovně (jako parametr se zadává IP adresa serveru, která je zjištěna automaticky pomocí dalšího skriptu):

```
java -cp .:sqlitejdbc-v056.jar QualityParser $IP
```

5.2.3 Quality Monitor

Část Quality Monitor získala svůj název podle jména třídy jádra výpočetní aplikace napsané v programovacím jazyce Java, která zpracovává algoritmus zjednodušeného E-modelu. V první řadě aplikace rozparsuje data uložená aplikací tshark a vybere pouze podstatné parametry potřebné k výpočtu či prezentaci výsledků. Jak již bylo zmíněno, všechna prezentovaná data jsou ukládána do SQLite 3 databáze, se kterou Java aplikace komunikuje pomocí open source API konektoru SQLiteJDBC, který je do aplikace importován ve formě jar balíčku.

5.3 Ovládání aplikace

Veškeré ovládání aplikace se provádí skrze webové uživatelské rozhraní pomocí tlačítek umístěných v horní části. Zde se také nachází indikátor stavu aplikace, tedy zda je samotný monitoring aktivní (Monitoring is running...) či je pozastaven (Monitoring is stopped.).

5.3.1 Start/Stop

V závislosti na stavu, ve kterém se aplikace nachází (zapnuto/vypnuto) je první tlačítko buď Start, nebo Stop. Jak již samotný název napovídá, slouží k samotné aktivaci monitoringu, popřípadě k jeho zastavení.

Kliknutím na tlačítko Start se pomocí Java skriptu a Ajaxu na straně serveru spustí php skript _start.php, který spustí na pozadí (bez blokace rozhraní, pomocí nohup) shell skript _start.sh. Tento skript aktivuje proces tsharku, který se vždy po 900 sekundách (15 minut) ukončí a získaná data z RTP filtru zapíše v textové podobě do souboru streams.log. Při tomto stavu nedochází vždy po každém ukončení tshark systémového procesu a uložení výsledků k výpočtu kvality a uložení do databáze, ale

vždy proběhnou 3 takovéto rotace, než dojde k automatickému výpočtu a soubor streams.log se vyčistí. Kliknutím na tlačítko Stop se pomocí Java skriptu a Ajaxu spustí skript `_stop.php`, který aktivuje shell skript `_stop.sh`, který má za úkol ukončit procesy tsharku a skriptu `_start.sh` pomocí příkazu kill.

5.3.2 Results

Tlačítko Results slouží k manuálnímu iniciování výpočtu pomocí E-modelu. Jak již bylo zmíněno, samotná aplikace vypočítá výsledky automaticky po 3 proběhnutích cyklu tsharku, tedy po 45 minutách. Tímto tlačítkem však můžeme výsledky nechat spočítat a zobrazit okamžitě. Tlačítko aktualizuje stránku, avšak proběhlo-li od poslední aktualizace vícero hovorů, je potřeba stránku aktualizovat později manuálně, aby se všechny výsledky zobrazily korektně.

Kliknutím na tlačítko Results se pomocí Java skriptu a Ajaxu na straně serveru spustí php skript `_perform.php`, který aktivuje shell skript `_perform.sh`. Pokud je aplikace v zapnutém stavu, tedy na pozadí běží proces tsharku, php skript nejprve spustí shell skript `_stop.sh`, stejně jako při ukončování monitoringu a až poté provede výpočet. Pokud byla aplikace před výpočtem v aktivním stavu a byla tedy ukončena stopovacím skriptem, je po provedení výpočtu opět aktivována spuštěním skriptu `_start.sh`, tedy stejně jako při zapínání aplikace. Shell skript `_perform.sh` v sobě obsahuje spuštění Java aplikace QualityParser, která zpracovává algoritmus zjednodušeného modelu a režii kolem parsování textového výstupu tsharku. Po provedení výpočtu skript vymaže obsah souboru `output.log`.

5.3.3 Refresh

Tlačítko Refresh má za úkol pouze obnovení stránky. Má obdobou funkci, jako klávesa F5 nebo tlačítko pro obnovení stránky ve webovém prohlížeči.

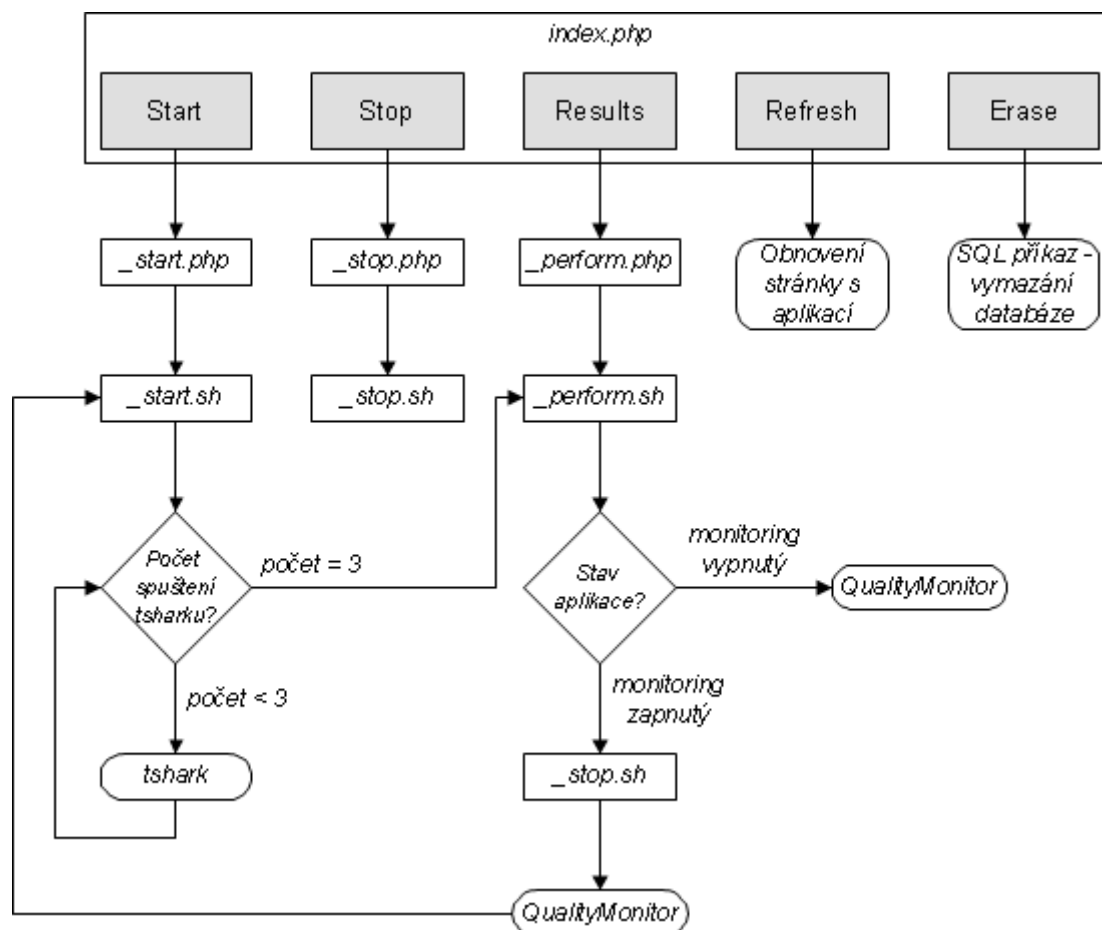
5.3.4 Erase

Tlačítko Erase má za úkol vymazání všech záznamů v databázi a tedy všech uložených a zobrazených výsledků. Po stisknutí tohoto tlačítka nedojde k automatickému obnovení stránky a prostředí, aplikaci je tedy nutno obnovit ručně, např. pomocí tlačítka Refresh.

Stiskem tlačítka Erase se pomocí Java skriptu a Ajaxu spustí skript `_erase.php`, který za pomoci SQL příkazu vymaže všechny záznamy v databázi.

5.4 Shrnutí stavu aplikace

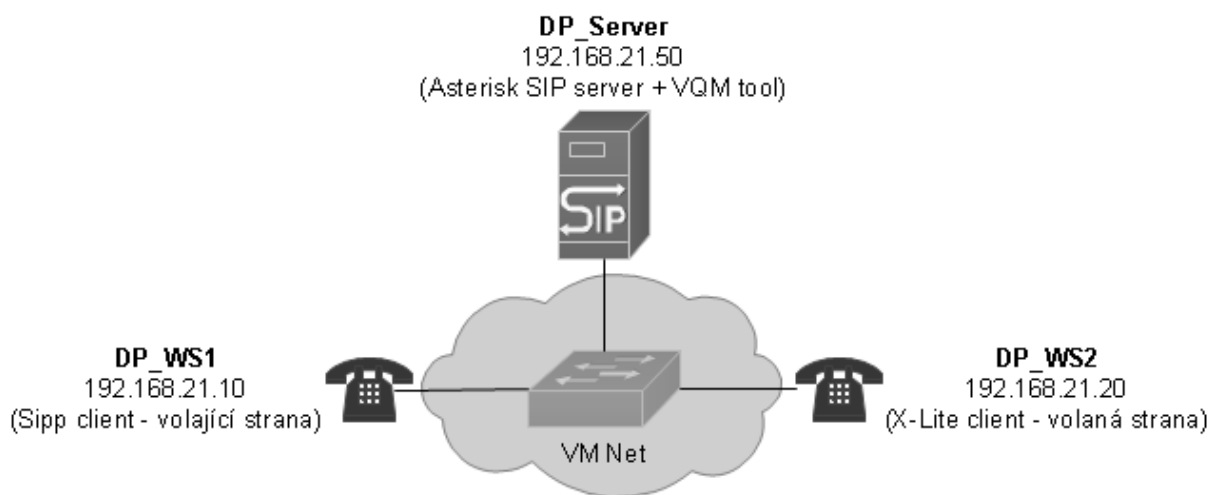
Nástroj pro monitorování kvality řeči ve VoIP, který je navržen v rámci této diplomové práce stačí k základnímu přehlednému monitorování kvality hlasu sestavených hovorů. Tato aplikace může sloužit jako pomůcka k řešení případných problémů, kdy si větší počet zákazníků stěžuje na dlouhodoběji sníženou kvalitu hovorů. V současném stavu aplikace zvládá podporu 4 základních kodeků, kterými jsou G.711 A-law, G.711 μ -law, G.729 a GSM 06.60 EFR. Pro kodeky jiné je aplikace transparentní a kvalitu nepočítá. Podrobná struktura ovládání aplikace včetně popisu spouštění jednotlivých procesů a skriptů je zobrazena na obrázku č. 5.3. V příloze G je uvedeno minimální aplikační vybavení nutné ke korektnímu provozu navržené aplikace.



Obr. 5.3 Struktura ovládání aplikace

6 Simulační a srovnávací měření

Další část praktického zpracování této diplomové práce se zabývá návrhem simulačního měření pro ověření věrohodnosti vytvořeného monitorovacího systému a srovnání s objektivní intrusivní metodou PESQ dle doporučení ITU-T P.862 [9]. Měření je dále založeno na vlivu ztrátovosti paketů, který má jeden z největších vlivů na kvalitu hlasu v IP telefonii. Vliv ztrátovosti je měřen v rozsahu 0-10 % s krokem 0,5 %. Na obrázku č. 6.1 je uvedeno schéma zapojení experimentálního simulačního pracoviště. Jelikož se jedná pouze o simulační měření, celé prostředí je virtuální a je vytvořeno za pomoci aplikace VMware.



Obr. 6.1 Schéma zapojení simulačního měření

6.1 Měření metodou E-model

Pro simulaci za použití metody E-modelu byla použita aplikace vytvořená za tímto účelem, která byla popsána v kapitole 5 této diplomové práce. Jak již bylo zmíněno, aplikace zpracovává zjednodušenou formu E-modelu, která vychází z doporučení ITU-T G.107 [3] a některé parametry výpočtu jsou nahrazeny konstantami dané ze statistického pozorování. Nad rámec doporučení je přidán vliv jakéhosi tandemového zapojení kodeků. Tedy detekuje-li aplikace, že každý účastník hovoru používá jiný kodek, nahradí výpočet parametru I_{e_ef} (viz rovnice (5.1)) následující relací:

$$I_{e_ef} = \frac{I_{e_ef_1} + I_{e_ef_2}}{2} \quad (6.1)$$

V podstatě jde tedy o nahrazení původního parametru I_{e_ef} průměrem jednotlivých kodeků a výsledný parametr kvality udává přehledovou kvalitu celého spojení [12], [17].

Pro generování hovorů na straně volajícího (virtuální stroj DP_WS1 s operačním systémem Linux – distribuce Ubuntu 10.11) jsem využil aplikace sipp (sip tester). Tato aplikace používá XML scénářů, dle kterých generuje celou strukturu hovoru, od registrace účastníka až po sestavení spojení, odeslání mediálních dat a ukončení spojení. Na obrázku č. 6.2 jde vidět právě probíhající spojení generované aplikací sipp. Spuštění aplikace sipp a vygenerování honoruje inicializováno z příkazové řádky operačního systému Ubuntu a struktura příkazu je následující:

```
sipp -aa -r 1 -t un -sf proxy.xml 192.168.21.50 -s 101 -i 192.168.21.10 -m 1
```

Nyní je pro představu dobré uvést význam jednotlivých parametrů příkazu:

- **-aa** – aktivuje automatické odesílání odpovědí typu 200 OK na požadavky INFO, UPDATE a NOTIFY.
- **-r 1** – četnost generování hovorů v počtech za sekundu. Pro naše účely je dostačující pouze jeden hovor.
- **-t un** – typ transportního protokolu. Parametr un udává protokol UDP, a vždy nový socket pro každé spojení.
- **-sf proxy.xml** – udává cestu k xml scénáři.
- **192.168.21.50** – IP adresa SIP serveru (stroje DP_Server).
- **-s 101** – uživatelské jméno účtu, který používáme pro registraci a tedy generování hovoru.
- **-i 192.168.21.10** – IP adresa odchozího rozhraní pro generování hovorů.
- **-m 1** – ukončení generování po dokončení určitého počtu hovorů, v našem případě tedy jednoho.

```
0 dead call msg (discarded)          0 out-of-call msg (discarded)
3 open sockets
158 Total RTP pkts sent                8.584 last period RTP rate (kB/s)

      Messages  Retrans  Timeout  Unexpected-Msg
REGISTER ----->      1      0      0
    100 <-----      0      0      0
    100 <-----      0      0      0
    200 <----- E-RTD1 1      0      0

INVITE ----->      1      0      0
    100 <-----      1      0      0
    180 <-----      1      0      0
    200 <----- E-RTD1 1      0      0

ACK ----->      1      0
  [ NOP ]
Pause [ 8000ms]      1
BYE ----->      0      0
    200 <-----      0      0      0

----- Waiting for active calls to end. Press [q] again to force exit. -----
```

Obr. 6.2 Průběh spojení v aplikaci sipp

Multimediální data jsou do scénáře aplikace sipp předávána ve formě PCAP souboru, tedy je nutno původní vzorek ve formátu WAV do tohoto formátu převést. K tomuto účelu jsem využil aplikace wav2rtp, kde se konverze provede jednoduchým příkazem v příkazové řádce operačního systému Linux:

```
wav2rtp -f input.wav -t output.pcap -c PCMU
```

Vysvětlení jednotlivých parametrů příkazu je následující:

- **-f input.wav** – vstupní soubor ve formátu WAV. Vzorek použitý v mých měřeních vychází přímo z doporučení ITU-T P.862 a jde tedy o standardizovaný měřicí vzorek který obsahuje vzorek hlasu i s kombinací ticha.
- **-t output.pcap** – výstupní soubor ve formátu PCAP, soubor, který pak používáme aplikací sipp.
- **-c PCMU** – kodek, který se použije pro konverzi do formátu PCAP. V mém případě měřím kodeky G.711 μ -law, G.711 A-law, G.729 a GSM.

V neposlední řadě je také dobré si ukázat strukturu xml scénáře, dle které aplikace sipp generuje hovory. Pro ukázkou uvádím strukturu žádosti INVITE:

```
<send retrans="500">
<![CDATA[
    INVITE sip:102@[remote_ip] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
```

```
From: [service] <sip:[service]@[remote_ip]>;tag=[call_number]
To: 102 <sip:102@[remote_ip]>
Call-ID: d///[call_id]
CSeq: 3 INVITE
User-Agent: Grandstream GXP2000 1.1.6.37
Contact:
<sip:[service]@[local_ip]:[local_port];transport=[transport]>
Max-Forwards: 70
Allow:
INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE,UPDATE,PRA
CK,MESSAGE
Supported: replaces, timer, path
Subject: Performance Test
Content-Type: application/sdp
Content-Length: [len]

v=0
o=[service] 8000 8001 IN IP[local_ip_type] [local_ip]
s=SIP Call
c=IN IP[media_ip_type] [media_ip]
t=0 0
m=audio [auto_media_port] RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=ptime:20
a=sendrecv
]]>

</send>
```

Přijímací část je opět tvořena virtuálním strojem (DP_WS2), na které jsou hovory přijímány s použitím aplikace X-Lite se zapnutou funkcí AA (auto anter), tedy k vyzvednutí hovoru dojde automaticky. Ukázka aplikace X-Lite je uvedena na obrázku 6.3.



Obr. 6.3 Ukázka aplikace X-Lite se zapnutou funkcí AA

Simulace síťového prostředí je provedena přímo aplikací VMware s využitím virtuálního L2 prostředí, tzv. VMnet, tedy není potřeba žádné další externí aplikace, která by virtuální stroje mezi sebou propojovala. Emulace ztrátovosti paketů v tomto prostředí je docíleno pomocí systému netem, kde aktivace náhodného zahazování některých paketů se provede následujícím příkazem, přímo z příkazové řádky operačního systému Linux:

```
tc qdisc add dev eth0 root netem loss 0.5%
```

6.2 Měření metodou PESQ

Pro měření metodou PESQ slouží stejná měřicí sestava jako v předchozí podkapitole. Navíc je na přijímací straně spuštěna aplikace Wireshark, kterou je zachytáván síťový provoz a tedy i přichází RTP pakety. Tyto RTP streamy lze pak přímo aplikací Wireshark uložit do souboru formátu WAV, který je pro měření metodou PESQ potřebný. Jak již bylo v popisu metody PESQ uvedeno (viz kapitola 4), ke stanovení kvality tímto způsobem, je zapotřebí i vzorek „původní“, tedy před přenesením. K tomuto účelu použijeme soubor ve formátu WAV, který konvertujeme do formátu PCAP pro použití aplikací sipp. Systém zpracovávající algoritmus PESQu vychází přímo z doporučení ITU-T P.862 [9], které obsahuje zdrojový kód aplikace v programovacím jazyce C++. K jejímu

zprovoznění je zapotřebí kód pouze zkompileovat a je připraven k použití. Srovnávání dvou vzorků, před a po přenesení, se pak opět provádí v příkazové řádce operačního systému Linux. Výstup výpočtu PESQu je uveden na obrázku č. 6.4 a struktura příkazu vypadá následovně:

```
./pesq +8000 reference.wav degraded.wav
```

Význam jednotlivých parametrů je následující:

- **+8000** – vzorkovací frekvence. Ve všech našich případech pracujeme s frekvencí 8 kHz.
- **reference.wav** – zvukový vzorek referenční, tedy před přenesením.
- **degraded.wav** – zvukový vzorek degradovaný, zachycený na přijímací straně aplikací Wireshark.

```
Reading reference file test_r.wav...done.  
Reading degraded file test_d.wav...done.  
Level normalization...  
IRS filtering...  
Variable delay compensation...  
Acoustic model processing...
```

```
Prediction : PESQ_MOS = 2.991
```

Obr. 6.4 Výstup aplikace pro výpočet metodou PESQ

6.3 Prezentace naměřených hodnot

Kvůli věrohodnosti měření, je pro každý kodek a krok ztrátovosti prováděno opakovaně 5 měření, aby bylo možno vyloučit možné chyby, které při simulaci vznikly. V této podkapitole uvedu výsledky pouze pro kodek G.711 A-law (viz tabulka 6.1 a obrázek 6.5) kvůli přehlednosti, naměřené výsledky pro kodeky ostatní jsou uvedeny v přílohách.

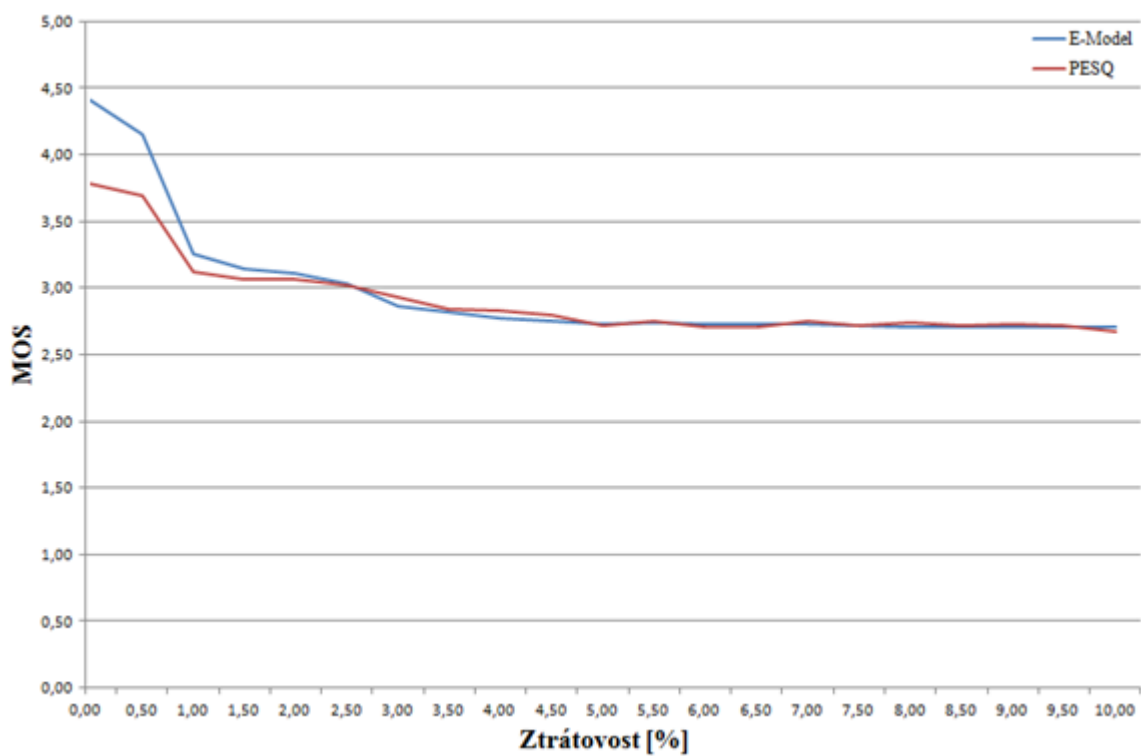
Tabulka 6.1: Naměřené výsledky pro kodek G.711 a-law

PL [%]	E-Model					Prům.	PESQ					Prům.
0	4,41	4,40	4,41	4,41	4,41	4,41	3,78	3,78	3,78	3,78	3,78	3,78
0,5	4,16	4,14	4,16	4,15	4,16	4,15	3,64	3,74	3,64	3,68	3,74	3,69
1	3,60	3,15	3,27	3,15	3,11	3,26	6,65	2,93	3,03	3,02	2,98	3,12
1,5	2,92	3,38	3,27	3,01	3,11	3,14	3,05	3,01	3,12	3,10	3,07	3,07
2	3,15	3,05	3,17	3,08	3,08	3,11	3,11	3,04	3,02	3,16	2,97	3,06
2,5	2,92	2,97	3,27	2,86	3,12	3,03	3,03	3,05	3,24	2,76	3,05	3,02

3	2,79	2,78	3,01	2,97	2,78	2,87	2,88	2,93	3,12	2,88	2,81	2,93
3,5	2,97	2,80	2,80	2,76	2,76	2,82	3,12	2,82	2,85	2,71	2,72	2,84
4	2,84	2,79	2,76	2,74	2,74	2,77	3,14	2,66	2,65	2,91	2,77	2,83
4,5	2,76	2,75	2,72	2,81	2,72	2,75	2,76	2,96	2,64	2,88	2,77	2,80
5	2,73	2,72	2,74	2,75	2,71	2,73	2,80	2,60	2,74	2,86	2,60	2,72
5,5	2,78	2,76	2,74	2,71	2,72	2,74	2,70	2,99	2,67	2,73	2,67	2,75
6	2,73	2,72	2,75	2,72	2,72	2,73	2,66	2,80	2,78	2,64	2,66	2,71
6,5	2,73	2,72	2,73	2,72	2,72	2,72	2,46	2,72	2,99	2,88	2,46	2,70
7	2,71	2,74	2,74	2,72	2,71	2,72	2,89	2,79	2,78	2,84	2,46	2,75
7,5	2,71	2,73	2,73	2,72	2,71	2,72	2,65	2,74	2,83	2,69	2,65	2,71
8	2,70	2,72	2,70	2,73	2,70	2,71	2,73	2,78	2,74	2,73	2,73	2,74
8,5	2,71	2,71	2,71	2,71	2,71	2,71	2,75	2,72	2,68	2,71	2,72	2,72
9	2,71	2,70	2,70	2,71	2,70	2,70	2,69	2,72	2,89	2,67	2,67	2,73
9,5	2,71	2,71	2,70	2,71	2,70	2,71	2,70	2,71	2,73	2,76	2,71	2,72
10	2,7	2,70	2,70	2,71	2,70	2,70	2,70	2,86	2,70	2,55	2,55	2,67

Tabulka 6.2:Legenda k tabulce č. 6.1

Zkratka	Význam
PL [%]	Ztrátovost paketů
E-Model	Hodnoty získané měřením navrženým systémem
PESQ	Hodnoty získané měřením metodou PESQ
Prům.	Průměr hodnot



Obr. 6.5 Grafická prezentace změřených hodnot pro kodek G.711 A-law

7 Zhodnocení výsledků a návrhy rozšíření

Poslední kapitola této diplomové práce se zabývá popisem a zhodnocením naměřených výsledků a návrhem možných vylepšení a rozšíření aplikace do budoucnosti, neboť tato aplikace je součástí systému vyvíjeného pod sdružením Cesnet.

7.1 Vyhodnocení naměřených výsledků

Všechny naměřené a získané výsledky jsou uvedeny v tabulkách a grafech. Pro kodek G.711 A-law v předchozí kapitole (viz tabulka 6.1 a obrázek 6.5) a pro kodeky ostatní, tedy G.711 μ -law, GSM 06.60 EFR a G.729 v příloze A až F.

Jak je z grafů na první pohled patrné, výsledky získané měřením užitím E-modelu zpracovaného do monitorovací aplikace, která je součástí praktické části této diplomové práce a výsledky získané měřením metodou PESQ přímo dle doporučení ITU-T P.862 [9] spolu velmi dobře, až na minimum odchylek korespondují. Pro lepší představu jsem na výsledky obou metod aplikoval korelační funkce dle následující relace:

$$r = \begin{cases} \frac{S_{XY}}{\sqrt{S_X^2 \cdot S_Y^2}} & \text{pro } S_X^2, S_Y^2 \neq 0 \\ 0 & \text{jinak} \end{cases} \quad (7.1)$$

kde jednotlivé parametry jsou definovány následovně:

$$S_X^2 = \frac{1}{n-1} \cdot \sum_{i=1}^n (X_i - \bar{X})^2 \quad (7.2)$$

$$S_Y^2 = \frac{1}{n-1} \cdot \sum_{i=1}^n (Y_i - \bar{Y})^2 \quad (7.3)$$

$$S_{XY} = \frac{1}{n-1} \cdot \sum_{i=1}^n (X_i - \bar{X}) \cdot (Y_i - \bar{Y}) = \frac{\sum_{i=1}^n X_i Y_i - n\bar{X}\bar{Y}}{\sqrt{(\sum_{i=1}^n X_i^2 - n\bar{X}^2) \cdot (\sum_{i=1}^n Y_i^2 - n\bar{Y}^2)}} \quad (7.4)$$

$$\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^n X_i \quad (7.5)$$

$$\bar{Y} = \frac{1}{n} \cdot \sum_{i=1}^n Y_i \quad (7.6)$$

Význam jednotlivých parametrů je následující – hodnoty \bar{X} a \bar{Y} udávají aritmetický průměr jednotlivých hodnot, tedy hodnot získaných metodou E-model a PESQ, hodnoty S_X^2 a S_Y^2 udávají rozptyl hodnot jednotlivých metod a v poslední řadě S_{XY} je parametr kovariance.

V tabulce číslo 7.1 jsou uvedeny míry korelace pro jednotlivé měřené kodeky. Shoda změřených výsledků neklesá v případě žádného kodeku pod 98%, je tedy patrné, že pro takového jednoduché použití je aplikace vhodná a výsledky korespondují se skutečnou kvalitou hovoru. Pro složitější a robustnější aplikace monitorovacího systému by bylo potřeba dlouhodobého a komplexnějšího testování při nasazení v „živé“ IP síti.

Tabulka 7.1: Míra korelace získaných výsledků pro měřené kodeky.

Kodek	Míra korelace [%]
G.711 A-law	98,95
G.711 μ -law	98,61
GSM	99,87
G.729	99,88

Z grafů je patrné, že největších nepřesností nabývají hodnoty v případě kodeků G.711 v rozmezí ztrátovosti 0 – 2 %. Toto je způsobeno tím, že výchozí hodnoty pro výpočet E-modelu, ze kterých vychází jeho zjednodušená forma, nekorespondují úplně přesně se skutečností při použití těchto kodeků a tak získaná hodnota parametru MOS je vyšší než teoretická maximální hodnota pro tyto kodeky. Toto však nepovažuji za vážný nedostatek, neboť v případě nízké ztrátovosti je u těchto kodeků kvalita nadprůměrná a za použití funkce PLC (Packet Loss Concealment), tedy maskování ztrátovosti se změna subjektivní kvality při takovýchto hodnotách ztrátovosti téměř nezmění [1], [10], [12], [16], [17].

7.2 Možné rozšíření aplikace

Jak již bylo zmíněno, aplikace je součástí základního monitorovacího systému v rámci projektu BESIP, řešeného na katedře telekomunikační techniky Vysoké školy báňské Technické univerzity Ostrava, proto v budoucnosti počítám s jeho rozšířením. Možné vylepšení aplikace jsem rozdělil do několika bodů:

- Podpora více kodeků – v současné době aplikace podporuje a umí rozpoznat kvalitu pouze u čtyř kodeků, které byly měřeny v rámci této diplomové práce. V budoucnu počítám s rozšířením o možnost stanovení kvality pro méně používané kodeky, což zaručí očekávané výsledky i za použití méně běžných VoIP klientů.

- Zahrnutí více parametrů do výpočtu – aplikace používá zjednodušeného E-modelu, kde je spousta původních parametrů nahrazena výchozími hodnotami dle doporučení. V budoucnosti vidím potenciál v zahrnutí pokročilejšího systému, který dokáže analyzovat IP prostředí, a tedy bude možno nahradit některé výchozí hodnoty hodnotami reálnými. Dále je možné se více zaměřit na využití kontrolního RTCP protokolu.
- Optimalizace grafického výstupu – optimalizací grafického výstupu je myšleno pokročilejší zobrazování výsledků, včetně možnosti vyhledávání, exportu do CSV formátu nebo statistické vyhodnocování včetně prezentace grafických výsledků ve zvoleném časovém intervalu.
- Optimalizace algoritmu – optimalizace algoritmu souvisí z velké části se zahrnutím více parametrů ve výpočtu E-modelu. Dalším možným vylepšením je zrychlení výpočtu a analýza spojení v Java aplikaci.

8 Závěr

Ve své diplomové práci jsem se zabýval hodnocením kvality řeči v IP prostředí, tedy technologii zvané VoIP. Svoji práci jsem rozdělil do několika stěžejních částí. První část práce (kapitola 2.) se zabývá obecným popisem VoIP technologie založené na signalizačním protokolu SIP, neboť na takovéto architektuře staví i praktická realizace mé diplomové práce. Cílem této kapitoly bylo čtenáři přiblížit používané technologie a základní strukturu SIP protokolu. Další kapitola (kapitola 3.) popisuje faktory mající největší dopad na kvalitu řeči v IP prostředí, z nichž byla vybrána ztrátovost paketů jako faktor, který je zkoumán v rámci simulačního měření v praktické části mé diplomové práce. Poslední kapitola teoretické části (kapitola 4.) se zabývá základním popisem metod pro hodnocení kvality řeči a popisem nejdůležitějších zástupců z řad metod jak subjektivních, tak objektivních.

V praktické části své diplomové práce jsem se zaměřil na vývoj jednoduchého monitorovacího systému pro hodnocení kvality řeči založeného na metodě E-model, která byla vybrána po dohodě s vedoucím mé práce a na základě předchozích zkušeností s touto metodou. E-model nabízí moderní přístup k výpočtu odhadované kvality, kterým je výpočetní model beroucí v potaz různé parametry komunikačního systému. V další části praktické realizace jsem vytvořil virtuální měřicí pracoviště, které mělo za úkol posoudit věrohodnost výsledků získaných vytvořeným monitorovacím systémem. Pro srovnávací měření byla zvolena metoda PESQ dle doporučení ITU-T P.862 [9], která je v dnešní době de-facto považována za referenční metodu pro stanovení MOS. Při měření jsem posuzoval měnící se kvalitu řeči v závislosti na projevené ztrátovosti paketů v IP prostředí.

Srovnávacím měřením jsem dokázal, že pro jednoduché aplikace je navržený monitorovací systém vhodný a dle přiložených grafů a parametrů korelace spolu hodnoty získané oběma metodami korespondovaly. Pro složitější aplikace na ústřednách s desítkami či stovkami současných hovorů by bylo nutné navržený systém podrobit komplexnějšímu a dlouhodobějšímu testování.

Cílem této práce bylo shrnout a popsat používané metody pro hodnocení kvality řeči hovorového signálu se zaměřením převážně na IP telefonii. Dále navrhnout a ověřit monitorovací systém, který by našel uplatnění při nasazení na open source pobočkové ústředně Asterisk. Vyvinutá aplikace je součástí základního monitorovacího balíčku použitého v projektu BESIP, řešeného na katedře telekomunikační techniky na fakultě elektrotechniky a informatiky VŠB-TUO a pod záštitou sdružení Cesnet, takže další vývoj aplikace stále není ukončen.

Řešení diplomové práce mi umožnilo seznámit se hlouběji s metodami hodnocení kvality řeči a podstatně rozšířilo můj odborný rozhled. Dále mi umožnilo seznámit se s vývojem aplikací pod

operačním systémem linux. Této oblasti bych se chtěl věnovat i nadále a na své zkušenosti popřípadě navázat doktorským studiem pod vedením doc. Ing. Miroslava Vozňáka, Ph.D.

Použitá literatura

- [1] VOZŇÁK, Miroslav. *Spojovací systémy*. 1. vyd. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2009, 196 s. ISBN 978-80-248-1961-7.
- [2] VOZŇÁK, Miroslav. *Voice over IP*. 1. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2008, 176 s. ISBN 978-80-248-1828-3.
- [3] ITU-T Recommendation G.107. *The E-model: A computational model for use in transmission planning*. Geneva: ITU-T, 04/2009.
- [4] ITU-T Recommendation G.108. *Application of the E-model: A planning guide*. Geneva: ITU-T, 09/1999.
- [5] ITU-T Recommendation G.109. *Definition of categories of speech transmission quality*. Geneva: ITU-T, 09/1999.
- [6] ITU-T Recommendation G.113. *Transmission impairments due to speech processing*. Geneva: ITU-T, 11/2007.
- [7] ITU-T Recommendation P.800. *Methods for subjective determination of transmission quality*. Geneva: ITU-T, 08/1996.
- [8] ITU-T Recommendation P.800.1. *Mean Opinion Score (MOS) terminology*. Geneva: ITU-T, 07/2006.
- [9] ITU-T Recommendation P.862. *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*. Geneva: ITU-T, 02/2001.
- [10] ITU-T Recommendation P.863. *Perceptual objective listening quality assessment*. Geneva: ITU-T, 2011.
- [11] MAHDI, Abdulhussain E. a Dorel PICOVICI. Advances in voice quality measurement in modern telecommunications. *Digital Signal Processing*[online]. 2009, roč. 19, č. 1, s. 79-103 [cit. 2012-05-01]. ISSN 10512004. DOI: 10.1016/j.dsp.2007.11.006. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1051200407001832>
- [12] TOMEŠ, Martin. *Hodnocení kvality hovoru E-modelem* [online]. Ostrava, 2010. Dostupné z: http://dspace.vsb.cz/bitstream/handle/10084/78804/TOM603_FEI_B2647_2612R059_2010.pdf?sequence=1. Bakalářská práce. VŠB-TUO. Vedoucí práce Ing. Miroslav Vozňák, Ph.D.

-
- [13] ŘEZÁČ, Filip. *Zabezpečená komunikace na SIP protokolu*. Ostrava, 2009. Diplomová práce. VŠB-TUO. Vedoucí práce Ing. Miroslav Vozňák, Ph.D.
- [14] ŘEZÁČ, Filip. *Uživatelská příručka k aplikaci SPITFILE*. Ostrava, 2009
- [15] PALAKAL, J. a M. J. ZORAN. Feature extraction from speech spectrograms using multi-layered network models. In: *Proc. IEEE International Workshop on Tools for Artificial Intelligence, Architectures, Languages and Algorithms*. 1989, IEEE, pp. 224-230.
- [16] VOZŇÁK, Miroslav. *Kvalita hlasu v prostředí VoIP*. V IT časopise Connect!, ISSN 1211-3085, 05/2009, Computer Press, pp. 14-17.
- [17] VOZŇÁK, M., TOMEŠ, M., VÁCLAVÍKOVÁ, Z., HALÁS, M., *E-model Improvement for Speech Quality Evaluation Including Codecs Tandeming*, V *Advances in Data Networks, Communications, Computers*, pp. 119-124, University of Algarve, Faro, Portugal, November 3-5, 2010, ISBN 978-960-474-245-5, ISSN 1792-6157
- [18] Diagnosing Voice Quality Impairments and Designing Solutions for Voice over IP Systems. In: *Dialogic* [online]. 04/2008 [cit. 2012-05-01]. Dostupné z: http://www.dialogic.com/~media/products/docs/media-server-software/8908_Des_Sol_for_VoIP_wp.pdf
- [19] RFC 3550. *RTP: A Transport Protocol for Real-Time Applications*. USA: IETF, 2003.
- [20] RFC 3611. *RTP Control Protocol Extended Reports (RTCP XR)*. USA: IETF, 2003.
- [21] RFC 3261. *SIP: Session Initiation Protocol*. USA: IETF, 2002.

Přílohy

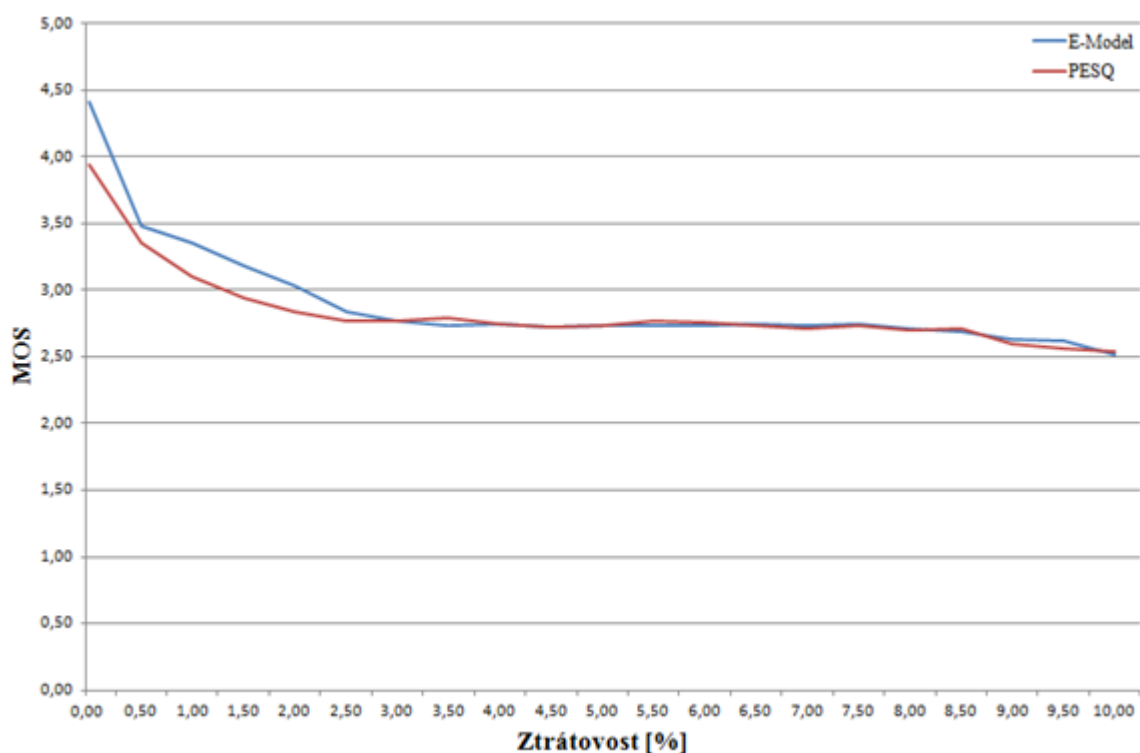
Seznam příloh

Příloha A: Tabulka naměřených hodnot pro kodek G.711 μ -law	iii
Příloha B: Grafická prezentace změřených hodnot pro kodek G.711 μ -law	iv
Příloha C: Tabulka naměřených hodnot pro kodek GSM 06.60 EFR	iv
Příloha D: Grafická prezentace změřených hodnot pro kodek GSM 06.60 EFR	v
Příloha E: Tabulka naměřených hodnot pro kodek G.729	v
Příloha F: Grafická prezentace změřených hodnot pro kodek G.729	vi
Příloha G: Minimální požadavky pro spuštění monitorovacího systému	vii

Příloha A: Tabulka naměřených hodnot pro kodek G.711 μ -law

PL [%]	E-Model					Prům.	PESQ					Prům.
0	4,41	4,41	4,41	4,41	4,41	4,41	4,01	3,89	3,97	3,93	3,91	3,94
0,5	3,38	3,56	3,41	3,38	3,65	3,48	3,35	3,41	3,32	3,34	3,37	3,36
1	3,32	3,38	3,48	3,33	3,29	3,36	3,05	3,17	3,11	3,07	3,13	3,11
1,5	3,31	3,24	3,01	3,01	3,35	3,18	2,89	3,02	2,91	2,97	2,94	2,95
2	3,05	2,98	3,01	3,04	3,08	„03	2,97	2,81	2,85	2,82	2,75	2,84
2,5	2,99	2,80	2,79	2,86	2,74	2,84	2,73	2,78	2,84	2,75	2,72	2,76
3	2,71	2,79	2,76	2,79	2,78	2,77	2,83	2,77	2,72	2,69	2,81	2,76
3,5	2,69	2,73	2,75	2,74	2,75	2,73	2,81	2,79	2,78	2,83	2,75	2,79
4	2,81	2,76	2,71	2,71	2,73	2,74	2,78	2,69	2,74	2,76	2,78	2,75
4,5	2,76	2,78	2,65	2,71	2,72	2,72	2,75	2,73	2,69	2,78	2,65	2,72
5	2,63	2,80	2,77	2,74	2,75	2,74	2,73	2,74	2,75	2,73	2,71	2,73
5,5	2,75	2,76	2,75	2,74	2,69	2,74	2,73	2,82	2,81	2,76	2,71	2,77
6	2,81	2,68	2,69	2,71	2,79	2,74	2,77	2,76	2,68	2,81	2,75	2,75
6,5	2,74	2,71	2,72	2,71	2,76	2,75	2,72	2,74	2,71	2,71	2,78	2,73
7	2,76	2,71	2,68	2,73	2,78	2,73	2,72	2,71	2,69	2,71	2,73	2,71
7,5	2,75	2,78	2,73	2,72	2,74	2,74	2,75	2,78	2,65	2,74	2,77	2,74
8	2,72	2,73	2,75	2,68	2,69	2,71	2,71	2,73	2,68	2,73	2,67	2,70
8,5	2,66	2,71	2,68	2,73	2,65	2,69	2,73	2,66	2,72	2,73	2,74	2,72
9	2,65	2,67	2,63	2,59	2,64	2,64	2,63	2,65	2,59	2,57	2,56	2,60
9,5	2,63	2,58	2,64	2,62	2,63	2,62	2,57	2,55	2,57	2,53	2,61	2,57
10	2,56	2,48	2,52	2,52	2,50	2,52	2,58	2,51	2,53	2,56	2,53	2,54

Příloha B: Grafická prezentace změřených hodnot pro kodek G.711 μ -law

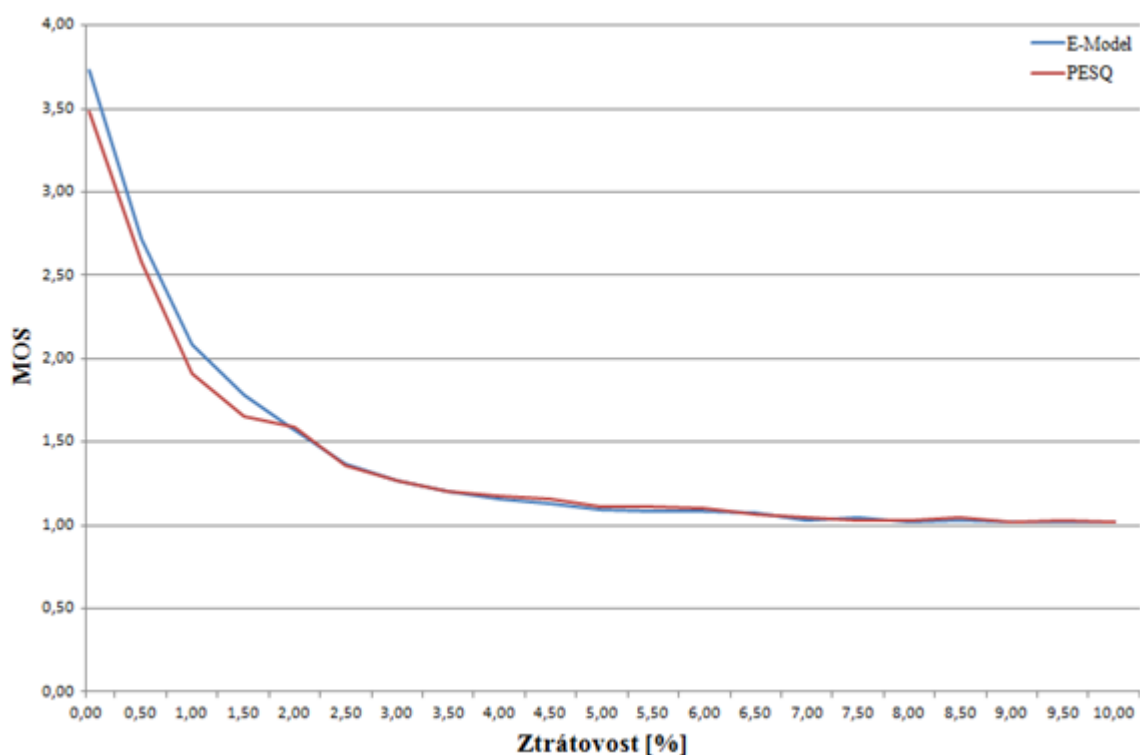


Příloha C: Tabulka naměřených hodnot pro kodek GSM 06.60 EFR

PL [%]	E-Model					Prům.	PESQ					Prům.
0	3,72	3,75	3,70	3,75	3,74	3,73	3,56	3,48	3,51	3,44	3,42	3,48
0,5	2,75	2,68	2,75	2,73	2,69	2,72	2,54	2,52	2,61	2,64	2,58	2,58
1	2,01	2,13	2,10	2,15	2,05	2,09	1,97	1,84	1,95	1,88	1,93	1,91
1,5	1,78	1,82	1,75	1,78	1,78	1,78	1,55	1,64	1,72	1,74	1,63	1,66
2	1,56	1,62	1,54	1,55	1,56	1,57	1,54	1,63	1,58	1,61	1,60	1,59
2,5	1,35	1,38	1,35	1,45	1,31	1,37	1,35	1,41	1,33	1,32	1,36	1,35
3	1,25	1,28	1,28	1,26	1,27	1,27	1,27	1,31	1,25	1,26	1,26	1,27
3,5	1,21	1,20	1,22	1,19	1,20	1,20	1,28	1,15	1,19	1,16	1,21	1,20
4	1,18	1,15	1,16	1,15	1,14	1,16	1,21	1,17	1,20	1,14	1,17	1,18
4,5	1,14	1,13	1,11	1,14	1,12	1,13	1,15	1,16	1,13	1,18	1,16	1,16
5	1,08	1,12	1,10	1,07	1,11	1,10	1,13	1,15	1,12	1,00	1,13	1,11
5,5	1,08	1,10	1,09	1,08	1,08	1,09	1,12	1,14	1,09	1,10	1,09	1,11
6	1,06	1,10	1,10	1,09	1,08	1,09	1,11	1,13	1,06	1,11	1,09	1,10
6,5	1,05	1,08	1,06	1,09	1,10	1,08	1,04	1,05	1,08	1,09	1,06	1,06
7	1,04	1,02	1,05	1,01	1,04	1,03	1,05	1,02	1,03	1,06	1,05	1,04

7,5	1,05	1,03	1,10	1,01	1,03	1,04	1,02	1,03	1,05	1,01	1,01	1,02
8	1,02	1,03	1,01	1,00	1,02	1,02	1,03	1,02	1,03	1,01	1,04	1,03
8,5	1,03	1,05	1,01	1,02	1,03	1,03	1,05	1,06	1,04	1,06	1,01	1,04
9	1,01	1,04	1,03	1,02	1,01	1,02	1,01	1,00	1,01	1,03	1,02	1,01
9,5	1,03	1,05	1,00	1,01	1,01	1,02	1,04	1,05	1,01	1,01	1,02	1,03
10	1,01	1,03	1,02	1,01	1,00	1,01	1,03	1,02	1,00	1,01	1,03	1,02

Příloha D: Grafická prezentace změřených hodnot pro kodek GSM 06.60 EFR

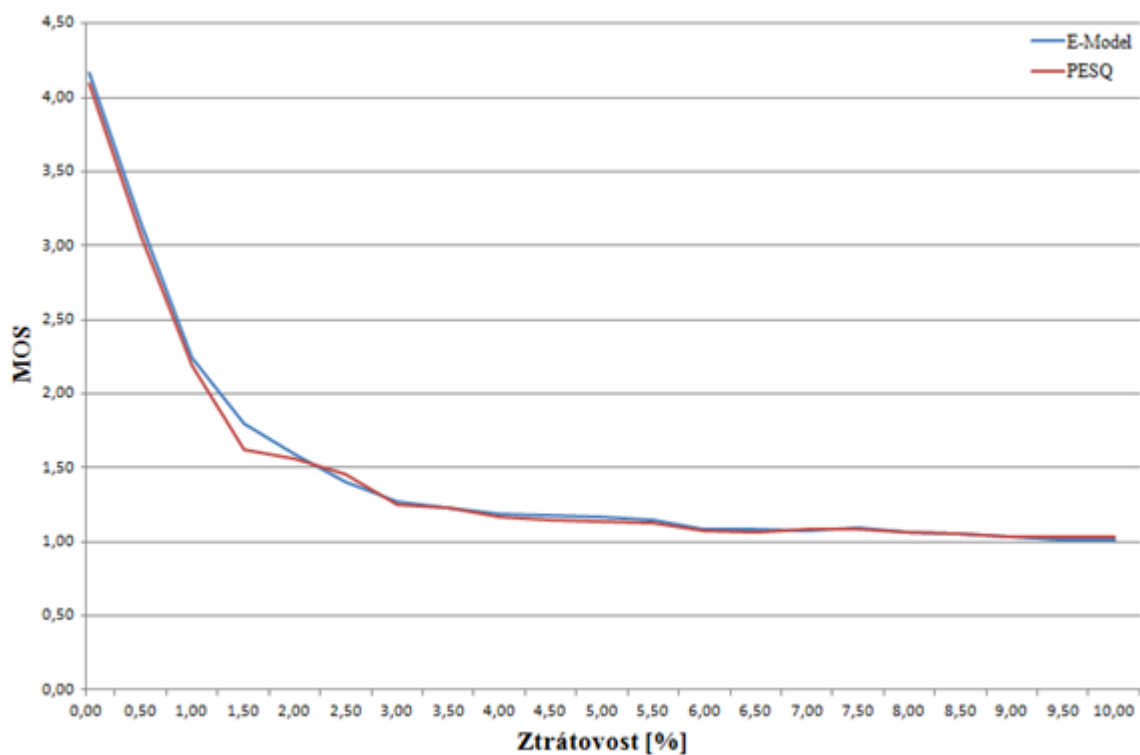


Příloha E: Tabulka naměřených hodnot pro kodek G.729

PL [%]	E-Model					Prům.	PESQ					Prům.
0	4,15	4,20	4,15	4,17	4,16	4,17	4,09	4,13	4,08	4,11	4,05	4,09
0,5	3,17	3,15	3,08	3,21	3,11	3,14	3,08	3,06	3,01	3,11	3,04	3,06
1	2,24	2,28	2,19	2,25	2,27	2,25	2,23	2,21	2,18	2,14	2,17	2,19
1,5	1,81	1,75	1,78	1,85	1,79	1,80	1,65	1,61	1,63	1,65	1,59	1,63
2	1,61	1,54	1,58	1,59	1,63	1,59	1,59	1,47	1,54	1,57	1,61	1,56
2,5	1,47	1,35	1,42	1,39	1,41	1,41	1,53	1,47	1,51	1,41	1,38	1,46
3	1,28	1,29	1,26	1,21	1,29	1,27	1,29	1,23	1,25	1,27	1,23	1,25
3,5	1,25	1,24	1,18	1,23	1,24	1,23	1,24	1,25	1,18	1,26	1,19	1,22

4	1,16	1,18	1,22	1,20	1,18	1,19	1,16	1,19	1,19	1,17	1,14	1,17
4,5	1,21	1,14	1,16	1,18	1,19	1,18	1,16	1,17	1,16	1,14	1,12	1,15
5	1,15	1,17	1,18	1,15	1,18	1,17	1,14	1,16	1,14	1,13	1,12	1,14
5,5	1,14	1,16	1,15	1,14	1,13	1,14	1,18	1,11	1,12	1,14	1,08	1,13
6	1,09	1,07	1,06	1,09	1,10	1,08	1,07	1,08	1,06	1,08	1,09	1,08
6,5	1,07	1,11	1,08	1,06	1,09	1,08	1,07	1,08	1,04	1,06	1,07	1,06
7	1,11	1,09	1,07	1,04	1,06	1,07	1,08	1,05	1,11	1,09	1,07	1,08
7,5	1,12	1,06	1,10	1,09	1,08	1,09	1,08	1,12	1,11	1,09	1,04	1,09
8	1,05	1,07	1,08	1,04	1,05	1,06	1,08	1,03	1,06	1,09	1,04	1,06
8,5	1,03	1,05	1,09	1,04	1,03	1,05	1,06	1,05	1,06	1,04	1,03	1,05
9	1,02	1,03	1,02	1,04	1,03	1,03	1,03	1,02	1,04	1,05	1,03	1,03
9,5	1,02	1,02	1,01	1,00	1,02	1,01	1,02	1,04	1,05	1,03	1,01	1,03
10	1,01	1,01	1,02	1,00	1,01	1,01	1,03	1,04	1,02	1,05	1,02	1,03

Příloha F: Grafická prezentace změřených hodnot pro kodek G.729



Příloha G: Minimální požadavky pro spuštění monitorovacího systému

Zde jsou uvedeny minimální požadavky na operační systém a aplikace, aby došlo ke korektnímu spuštění a provozu navrženého monitorovacího systému.

Požadavky na operační systém:

- Ubuntu 11.10

Požadavky na aplikační vybavení:

- Wireshark v. 1.6.2-1
- Lighttpd v. 1.4.28-2ubuntu2.1 s php-cgi modulem
- PHP
- Java6 Runtime Environment
- SQLite3 v. 3.7.7-2ubuntu2
- Nohup